PATENT ABSTRACTS OF JAPAN

(11)Publication number : **2002-049575**
(43)Date of publication of application : **15.02.2002**

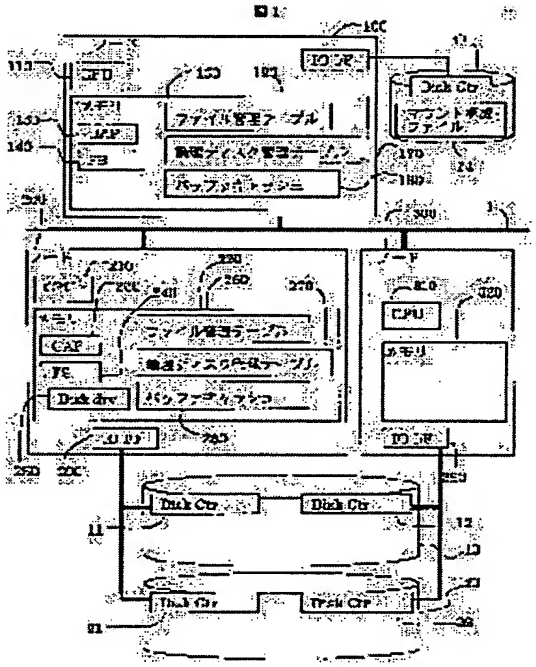
(51)Int.Cl. **G06F 13/14**
G06F 3/06
G06F 12/00
G06F 12/16
G06F 13/00

(21)Application number : **2000-233291**
(22)Date of filing : **01.08.2000**

(71)Applicant : **HITACHI LTD**
(72)Inventor : **ITO AKIHIRO**
UTSUNOMIYA NAOKI
SONODA KOJI
KUMAZAKI HIROYUKI

(54) FILE SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a file system which can shorten the time needed for IO path switching and conceal IO path switching processing from general users.
SOLUTION: In this system which defines file IDs for every files, a file server FS refers to a file management table and finds a logical disk ID for accessing a file when a user application UAP makes a request to access while specifying the file ID of the file. Further, the file server refers to a logical disk management table to find the IO path corresponding to the logical disk ID and uses the IO path to access a physical disk unit. If trouble occurs to the IO path of an in-operation system, the logical disk management tables of all nodes are rewritten to switch the IO path.



LEGAL STATUS

[Date of request for examination] 18.07.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－49575
（P2002－49575A）
(43)公開日　平成14年２月15日(2002.2.15)

(51)Int.Cl.⁷	識別記号	ＦＩ		テーマコード（参考）
Ｇ０６Ｆ　13/14	３１０	Ｇ０６Ｆ　13/14	３１０Ｈ	５Ｂ０１４
3/06	３０４	3/06	３０４Ｂ	５Ｂ０１８
12/00	５１４	12/00	５１４Ｅ	５Ｂ０６５
12/16	３１０	12/16	３１０Ａ	５Ｂ０８２
13/00	３０１	13/00	３０１Ｐ	５Ｂ０８３

審査請求　未請求　請求項の数20　ＯＬ　（全 26 頁）

(21)出願番号　特願2000－233291(P2000－233291)

(22)出願日　平成12年８月１日(2000.8.1)

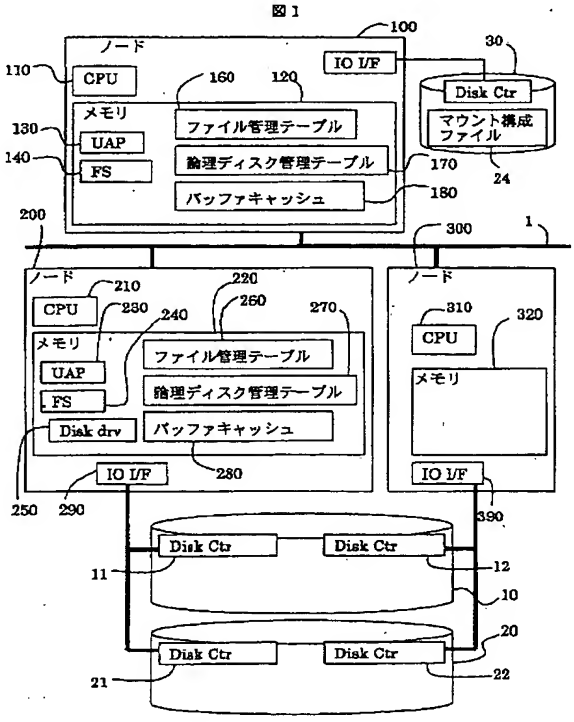
(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　伊藤　昭博
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(72)発明者　宇都宮　直樹
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(74)代理人　100078134
弁理士　武　顕次郎

最終頁に続く

(54)【発明の名称】　ファイルシステム

(57)【要約】
【課題】　ＩＯパス切り替えのために要する時間を短縮し、一般ユーザからＩＯパス切り替え処理を隠蔽することのできるファイルシステム。
【解決手段】　ファイル毎にファイルＩＤが定義されているシステムにおいて、ユーザアプリケーションＵＡＰからのファイルＩＤを指定したアクセス要求に対して、ファイルサーバＦＳはファイル管理テーブルを参照し、そのファイルをアクセスするための論理ディスクＩＤを求める。ファイルサーバは、さらに、論理ディスク管理テーブルを参照し、論理ディスクＩＤに対応するＩＯパスを求め、そのＩＯパスを使って物理ディスク装置にアクセスする。運用系のＩＯパスに障害発生時、全ノードの論理ディスク管理テーブルを書き換えることによってＩＯパスの切り替えを行う。

図１



【特許請求の範囲】

【請求項１】　ファイル毎にファイルＩＤが定義されており、複数の物理ディスク装置に分散管理されたファイルの処理を行う１または複数のファイルサーバを有するファイルシステムにおいて、ファイルＩＤ及び該ファイルＩＤに対応するファイルが格納されている論理ディスクの論理ディスクＩＤを含むファイル管理テーブルと、論理ディスクＩＤ及び前記論理ディスクに対応する１つ以上の物理ディスク装置にアクセスするための１つ以上のＩＯパスを含む論理ディスク管理テーブルとを備え、ユーザからのファイルＩＤを指定したファイルへのアクセス要求を受信したファイルサーバは、ファイル管理テーブルを参照し、前記ファイルＩＤから前記ファイルが格納されている論理ディスクの論理ディスクＩＤを決定し、論理ディスク管理テーブルを参照して前記論理ディスクＩＤから前記論理ディスクに対応する物理ディスク装置にアクセスするためのＩＯパスを決定し、決定したＩＯパスを使用して物理ディスク装置にアクセスすることを特徴とするファイルシステム。

【請求項２】　ネットワークに接続されたそれぞれの内部にファイルサーバが構成された複数のノードと、複数のノードの少なくとも２つのノードに共通に接続された物理ディスク装置とを備え、ファイル毎にファイルＩＤが定義されており、前記複数の物理ディスク装置に分散管理されたファイルの処理を行うファイルシステムにおいて、複数のノードのそれぞれは、ファイルＩＤ及び前記ファイルＩＤに対応するファイルが格納されている論理ディスクの論理ディスクＩＤを含むファイル管理テーブルと、論理ディスクＩＤ及び前記論理ディスクに対応する１つ以上の物理ディスク装置にアクセスするための１つ以上のＩＯパスを含む論理ディスク管理テーブルとを備え、ユーザからのファイルＩＤを指定したファイルへのアクセス要求を受信したファイルサーバは、ファイル管理テーブルを参照し、前記ファイルＩＤから前記ファイルが格納されている論理ディスクの論理ディスクＩＤを決定し、論理ディスク管理テーブルを参照して前記論理ディスクＩＤから前記論理ディスクに対応する物理ディスク装置にアクセスするためのＩＯパスを決定し、決定したＩＯパスを使用して物理ディスク装置にアクセスすることを特徴とするファイルシステム。

【請求項３】　前記ＩＯパスを特定する情報は、ノード番号、ＩＯインターフェイス番号及びディスクコントローラ番号からなることを特徴とする請求項２記載のファイルシステム。

【請求項４】　前記ファイルＩＤから決定した論理ディスクＩＤに対応する物理ディスク装置が、他のノードであるリモートノードに接続されている場合、自ノードのファイルサーバは、前記リモートノードにアクセス要求を送信し、前記アクセス要求を受信した前記リモートノードのファイルサーバが前記物理ディスク装置に格納された該当ファイルにアクセスすることを特徴とする請求項３記載のファイルシステム。

【請求項５】　ネットワークに接続されたそれぞれの内部にファイルサーバが構成された複数のノードと、複数のノードの少なくとも２つのノードに共通に接続された物理ディスク装置とを備え、ファイル毎にファイルＩＤが定義されており、前記複数の物理ディスク装置に分散管理されたファイルの処理を行うファイルシステムにおいて、前記物理ディスク装置の少なくとも１つは、１つのマウントポイントに対して物理ディスク装置にアクセスするための１つ以上のＩＯパスを対応づける情報を１つのエントリに含むマウント構成ファイルを格納しており、システム立ち上げ時、前記マウント構成ファイルを格納するディスク装置が接続されたノードのファイルサーバは、前記マウント構成ファイルを読み出し、前記マウント構成ファイルの１つのエントリに記載された１つ以上のＩＯパスに対して１つの論理ディスクＩＤを自動設定し、前記論理ディスクＩＤと前記ＩＯパスとの対応関係を論理ディスク管理テーブルに登録し、他の全てのノードのファイルサーバと通信を行うことによって、前記論理ディスク管理テーブルの内容を全てのノードの論理ディスク管理テーブルに複写し、前記マウント構成ファイルによって前記ＩＯパスに対応づけられたマウントポイントに前記論理ディスクＩＤに対応する論理ディスクをマウントし、複数のノードのそれぞれは、ファイルＩＤ及び前記ファイルＩＤに対応するファイルが格納されている論理ディスクの論理ディスクＩＤを含むファイル管理テーブルと、論理ディスクＩＤ及び前記論理ディスクに対応する１つ以上の物理ディスク装置にアクセスするための１つ以上のＩＯパスを含む論理ディスク管理テーブルとを備え、ユーザからのファイルＩＤを指定したファイルへのアクセス要求を受信したファイルサーバは、ファイル管理テーブルを参照し、前記ファイルＩＤから前記ファイルが格納されている論理ディスクの論理ディスクＩＤを決定し、論理ディスク管理テーブルを参照して前記論理ディスクＩＤから前記論理ディスクに対応する物理ディスク装置にアクセスするためのＩＯパスを決定し、決定したＩＯパスを使用して物理ディスク装置にアクセスすることを特徴とするファイルシステム。

【請求項６】　前記マウント構成ファイルは、ＩＯパス毎に前記ＩＯパスが使用できるか否かを登録する使用可否情報を含み、論理ディスク管理テーブルは、前記論理ディスク管理テーブルに登録されているＩＯパス毎に稼働状態を保持する状態フラグを含み、マウント処理を行うファイルサーバは、システム立ち上げ時に、前記マウント構成ファイルの１つのエントリに記載された複数のＩＯパスのうち、前記マウント構成ファイルの使用可否情報に「使用可」と登録されたＩＯパスの１つについて、論理ディスク管理テーブルの前記ＩＯパスに対応する状態フラグに「使用中」状態と登録し、前記マウント

構成ファイルの使用可否情報に「使用可」と登録された残りのＩＯパスについて、前記論理ディスク管理テーブルの前記ＩＯパスに対応する状態フラグに「待機中」状態と登録し、前記マウント構成ファイルの使用可否情報に「使用不可」と登録されたＩＯパスについて、前記論理ディスク管理テーブルの前記ＩＯパスに対応する状態フラグに「使用不可」状態と登録し、各ノードのファイルサーバは、通常運用時、前記論理ディスク管理テーブルの状態フラグが「使用中」状態となっている運用系のＩＯパスを用いて、物理ディスク装置にアクセスすることを特徴とする請求項５記載のファイルシステム。

【請求項７】　前記物理ディスク装置のディスクコントローラ、前記物理ディスク装置が接続されたノードのＩＯインターフェイスなどの障害によって、運用系ＩＯパスが使用不可能になったとき、前記障害を検出したノードのファイルサーバは、前記ノードの論理ディスク管理テーブルを更新し、前記使用不可能になったＩＯパスの状態フラグを「使用不可」とし、前記使用不可能になったＩＯパスと同じ論理ディスクＩＤに対応付けられているＩＯパスのうち状態フラグが「待機中」である１つのＩＯパスの状態フラグを「使用中」として新運用系ＩＯパスとした後、全ての他のリモートノードのファイルサーバと通信を行い、前記論理ディスク管理テーブルの内容を全ノードの論理ディスク管理テーブルに複写することによって、前記物理ディスク装置へアクセスするためのＩＯパスを前記使用不可能となったＩＯパスから前記新運用系ＩＯパスに切り替えることを特徴とする請求項６記載のファイルシステム。

【請求項８】　前記ＩＯパスの切り替え処理の間、使用不可能となったＩＯパスに含まれるノードのファイルサーバは、使用不可能になったＩＯパスへのアクセス要求を保留し、ＩＯパスの切り替え処理終了時、保留していた前記アクセス要求を新運用系ＩＯパスに含まれるノードに転送することを特徴とする請求項７記載のファイルシステム。

【請求項９】　前記ＩＯパスの切り替え処理の間、使用不可能となったＩＯパスに含まれるノードにアクセス要求を発行したファイルサーバは、前記アクセス要求がタイムアウトになった場合、論理ディスク管理テーブルを参照し論理ディスクＩＤからＩＯパスを求め直し、新しく求め直したＩＯパスを使用して、物理ディスク装置にアクセスし直すことを特徴とする請求項７記載のファイルシステム。

【請求項１０】　前記複数のノードのそれぞれは、物理ディスク装置との間に転送されるデータを一時的に保持するバッファキャッシュを備え、ＩＯパスの切り替え処理時、使用不可能になったＩＯパスに含まれるノードのファイルサーバと、新運用系ＩＯパスに含まれるノードのファイルサーバとが通信を行い、前記使用不可能になったＩＯパスに含まれるノードの主記憶内に存在し、物理ディスク装置に書き戻す必要があるバッファキャッシュ及びファイル管理テーブルを前記新運用系ＩＯパスに含まれるノードに転送することを特徴とする請求項７記載のファイルシステム。

【請求項１１】　前記物理ディスク装置内のディスクコントローラは、ディスク領域との間で転送されるデータを一時的に保持するディスクキャッシュを備え、前記物理ディスク装置内の別のディスクコントローラが備えるディスクキャッシュに格納されたデータをディスク領域に書き戻す機能を有し、ＩＯパスの切り替え処理時、前記使用不可能になったＩＯパスを使ってアクセスしていた物理ディスク装置内に設けられた前記使用不可能になったＩＯパスに含まれるディスクコントローラが備えるディスクキャッシュに格納されたデータのうち、前記物理ディスク装置に書き戻す必要のあるデータを、前記物理ディスク装置内に存在し、新運用系ＩＯパスに含まれるディスクコントローラを使用して、前記物理ディスク装置に書き戻すことを特徴とする請求項７記載のファイルシステム。

【請求項１２】　ＩＯパスの切り替え終了時、マウント構成ファイルを格納するディスク装置が接続されたノードのファイルサーバが前記マウント構成ファイルを更新し、前記使用不可能となったＩＯパスの使用可否情報を「使用不可」に書き換えることを特徴とする請求項７記載のファイルシステム。

【請求項１３】　使用不可能となっていたＩＯパスが再び使用できるようになったとき、前記複数のノードのある１つのノードのファイルサーバが、自ノードの論理ディスク管理テーブルに登録された前記ＩＯパスの状態フラグを「使用不可」状態から「待機中」状態に更新し、前記ファイルサーバが他の全てのノードのファイルサーバと通信を行うことにより、全てのノードの論理ディスク管理テーブルに前記更新内容を複写した後、マウント構成ファイルを格納するディスク装置が接続されたノードのファイルサーバが、前記マウント構成ファイルに登録された前記ＩＯパスの使用可否情報を「使用可」に書き換えることにより、前記ＩＯパスを待機系ＩＯパスとしてシステムに復旧させることを特徴とする請求項７記載のファイルシステム。

【請求項１４】　物理ディスク装置が接続されたノードに障害が発生したとき、前記ノードの障害を検出した他のノードのファイルサーバは、自ノードの論理ディスク管理テーブルを検索し、障害発生ノード番号から障害発生ＩＯパス及び前記障害発生ＩＯパスと同じ論理ディスクＩＤに対応付けられているＩＯパスのうち状態フラグが「待機中」であるＩＯパスの１つを新運用系ＩＯパスとして求め、この新運用系ＩＯパスに含まれるノードのファイルサーバにＩＯパスの切り替え処理を行うように要求し、前記要求を受けた前記ファイルサーバは、自ノードの論理ディスク管理テーブルを更新し、前記障害発

生ＩＯパスの状態フラグを「使用不可」とし、前記新運用系ＩＯパスの状態フラグを「使用中」とした後、他の全てのノードのファイルサーバと通信を行い、前記論理ディスク管理テーブルの内容を全ノードの論理ディスク管理テーブルに複写することによって、前記物理ディスク装置へアクセスするためのＩＯパスを前記障害発生ＩＯパスから前記新運用系ＩＯパスに切り替えることを特徴とする請求項６記載のファイルシステム。

【請求項１５】　ＩＯパスの切り替え処理の間、前記障害発生ＩＯパスに含まれるノードにアクセス要求を発行したファイルサーバは、前記アクセス要求がタイムアウトになった場合、論理ディスク管理テーブルを参照し論理ディスクＩＤからＩＯパスを求め直し、新しく求め直したＩＯパスを使用して、物理ディスク装置にアクセスし直すことを特徴とする請求項１４記載のファイルシステム。

【請求項１６】　前記物理ディスク装置が接続されているノードは、自ノードの状態にかかわりなく自ノードが備えるメモリ内のデータを読み出し、読み出したデータを他のノードに転送する機能を持ったハードウェアを有し、ＩＯパスの切り替え処理時、前記ハードウェアを用いて、前記障害発生ＩＯパスに含まれるノードの主記憶内に存在し、物理ディスク装置に書き戻す必要があるバッファキャッシュ及びファイル管理テーブルを前記新運用系ＩＯパスに含まれるノードに転送することを特徴とする請求項１４記載のファイルシステム。

【請求項１７】　前記物理ディスク装置内のディスクコントローラは、ディスク領域との間で転送されるデータを一時的に保持するディスクキャッシュを備え、前記物理ディスク装置内の別のディスクコントローラが備えるディスクキャッシュに格納されたデータをディスク領域に書き戻す機能を有し、ＩＯパスの切り替え処理時、前記障害発生ＩＯパスを使ってアクセスしていた物理ディスク装置内に設けられた前記障害発生ＩＯパスに含まれるディスクコントローラが備えるディスクキャッシュに格納されたデータのうち、前記物理ディスク装置に書き戻す必要のあるデータを、前記物理ディスク装置内に存在し、新運用系ＩＯパスに含まれるディスクコントローラを使用して、前記物理ディスク装置に書き戻すことを特徴とする請求項１４記載のファイルシステム。

【請求項１８】　ＩＯパスの切り替え終了時、マウント構成ファイルを格納するディスク装置が接続されたノードのファイルサーバが前記マウント構成ファイルを更新し、使用できなくなった運用系ＩＯパスの使用可否情報を「使用不可」に書き換えることを特徴とする請求項１４記載のファイルシステム。

【請求項１９】　前記マウント構成ファイルは、ＩＯパス毎に前記ＩＯパスが使用できるか否かを登録する使用可否情報を含み、前記論理ディスク管理テーブルは、該論理ディスク管理テーブルに登録されているＩＯパス毎に稼働状態を保持する状態フラグを含み、前記マウント処理を行うファイルサーバは、システム立ち上げ時に、前記マウント構成ファイルの使用可否情報に「使用可」と登録されたＩＯパスについて、論理ディスク管理テーブルの前記ＩＯパスに対応する状態フラグに「使用中」状態と登録し、前記マウント構成ファイルの使用可否情報に「使用不可」と登録されたＩＯパスについて、論理ディスク管理テーブルの前記ＩＯパスに対応する状態フラグに「使用不可」状態と登録し、通常運用時、ファイルサーバは、前記論理ディスク管理テーブルの状態フラグが「使用中」状態のＩＯパスからアクセスされる物理ディスク装置にファイルをミラーリングすることを特徴とする請求項５記載のファイルシステム。

【請求項２０】　前記使用中ＩＯパスの１つに障害が発生したとき、この障害を検出したノードのファイルサーバは、自ノードの論理ディスク管理テーブルを更新し、障害が発生した前記ＩＯパスの状態フラグを「使用不可」とした後、他の全てのノードのファイルサーバと通信を行い、前記論理ディスク管理テーブルの内容を全ノードの論理ディスク管理テーブルに複写し、マウント構成ファイルを格納するディスク装置が接続されたノードのファイルサーバが、前記マウント構成ファイルを更新し、前記障害が発生したＩＯパスの使用可否情報を「使用不可」に書き換えることによって、障害が発生したＩＯパスを切り放すことを特徴とする請求項１９記載のファイルシステム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、複数のディスク装置に分散管理されたファイルの処理を行うファイルシステムに係り、特に、１つのディスク装置へアクセスするためのＩＯパスが複数存在する場合に、ＩＯパスの切り替えを制御を行って一方のパスからディスク装置へアクセスすることができるファイルシステムに関する。

【０００２】

【従来の技術】従来技術によるファイルシステムの１つであるＵＮＩＸ（登録商標）ファイルシステムは、各ファイル毎にユニークに決まる番号（ファイルＩＤ）が定義されており、ファイルサーバがファイルＩＤを指定することによって、リード・ライト処理を行うファイルを特定することができる。そして、ファイルサーバは、ファイルＩＤとそのファイルが格納されているディスク装置にアクセスするためのＩＯパス（ＩＯパスを決定する情報は、ノード番号、ＩＯインターフェイス番号、装置番号などである）との対応関係をメモリ上のファイル管理テーブル（ＵＮＩＸではｉｎｏｄｅと呼ばれる）に登録して管理している。この管理方法については、例えば、(The Design of The Unix Operating System; Maurice J. Bach; p60-p72)に述べられている。

【０００３】ファイルＩＤを指定したリード・ライトア

クセス要求に対して、ファイルサーバは、前述のファイル管理テーブルを参照し、ファイルＩＤからディスク装置にアクセスするためのＩＯパス名を決定し、そのＩＯパスを用いてディスク装置にアクセスを行う。ファイル管理テーブルには、ＩＯパス情報の他に、ファイルサイズやファイルの更新日付などのファイル管理情報が登録されており、このファイル管理情報は、ファイルがオープンされたとき、ディスク装置から読み出され、定期的あるいはファイルをクローズしたときに、ディスク装置に書き戻される。ユーザがファイルにアクセスするとき指定するファイル名からファイルＩＤへの変換は、ファイルサーバが行っている。

【０００４】また、複数のディスク装置をシステムで取り扱う場合、あるディスク装置Ａで管理されるディレクトリネームツリー内のいずれかのディレクトリ、例えば、Ｘに別のディスク装置Ｂで管理されるネームツリーを組み込むという操作によって、複数のディスク装置を１つのネームツリー内に見せるという方法が知られている。この方法によれば、ユーザは、ディレクトリＸにアクセスすればディスク装置Ｂ内のファイルにアクセスすることができる。この方法は、マウント処理と呼ばれているものである。ファイルサーバは、起動時にある特定のディスク装置（ルートデバイス）を起点として前述したマウント処理を次々に行い、ユーザには複数のディスク装置を１つのネームツリーとして見せるようにしている。この起動時におけるマウント処理を行うためのディスク装置とネームツリー上のディレクトリ名（マウントポイント）との対応関係を記述した情報は、ルートデバイスにマウント構成ファイルとして記録されており、ファイルサーバは、起動時にこのマウント構成ファイルに記載された情報に従ってマウント処理を行う。

【０００５】マウント構成ファイルには、ディスク装置を特定する情報として、そのディスク装置にアクセスするためのＩＯパスの情報が記載されている。ファイルサーバは、マウント処理の実行時に、マウント構成ファイルに記載されたＩＯパスとマウントポイントとの対応関係をメモリ上のマウント構成情報に読み込む。そして、ファイルサーバは、ユーザがファイル名を指定してファイルをオープンするとき、前述のマウント構成情報を元にファイルが格納されている物理ディスク装置にアクセスするためのＩＯパスを求め、ファイル管理テーブルを作成する。従って、システム管理者は、システムに新しいディスク装置を接続するなどしてシステムの構成を変更したとき、マウント構成ファイルを書き換えることによって、新しい構成情報を計算機システムに設定する必要がある。

【０００６】一方、計算機システムの信頼性を向上させるため、異なる２つのノードを１つのディスク装置に物理的に接続し、異なる２通りのＩＯパスからディスク装置にアクセスすることができる構成にしておき、通常の運用時に一方のＩＯパスを使用し、ノード障害が発生して使用中のＩＯパスが使用できなくなったとき、もう一方のＩＯパスを用いて別のノードからディスク装置にアクセスするようにすることによって、障害発生時においてもディスク装置の可用性（アベイラビリティ）を保つ方法が、例えば、特開平１０－２７５０９０号公報等に記載されて知られている。

【０００７】また、ディスク装置の信頼性を向上するために、ファイルを複数のディスクに多重化して記録する方法（ミラーリング）がよく知られている。ミラーリングを行う場合、一般に、論理ボリュームという概念が用いられる。ミラーリングは、複数の物理ディスク装置を、１つの論理ボリュームとしてユーザに見せる仕組みである。ユーザは、予め複数の物理ディスク装置の情報を登録した「論理ボリューム」を作成しておく。そして、ユーザがこの論理ボリュームに対して、物理ディスク装置と同様にアクセスすると、複数の物理ディスクへのファイルのミラーリングが行われる。論理ボリュームを使用することにより、ファイルを複数のディスク装置に分散記録するストライピングを行うことも可能となる。

【０００８】

【発明が解決しようとする課題】前述で説明した使用中のＩＯパスが使用不可能になったとき、物理ディスク装置にアクセスするためのＩＯパスを別のＩＯパスに切り替える処理を、従来のＵＮＩＸのファイルシステムに適用して動的に行おうとすると、ファイル管理テーブル及びマウント構成情報を検索し、使用できなくなったＩＯパス名を新しいＩＯパス名に書き換える操作を行う必要がある。前述のファイル管理テーブルのエントリを書き換える処理は、オープンされているファイルの個数だけ全てについて行わなければならない。この結果、従来のＵＮＩＸのファイルシステムに、前述したＩＯパスの切り替えの技術を適用した場合、ファイル管理テーブルのエントリを書き換える処理に時間がかかり、その間その物理ディスク装置にＩＯ処理を行うことができないという問題点を生じることになる。

【０００９】また、ＩＯパスに障害が発生したときに、単純にＩＯパスを切り替えるだけでは、障害発生前に物理ディスク装置にアクセスを行っていたノードが持っていたバッファキャッシュ（物理ディスク装置にリード・ライトするときにデータを一時的に蓄えておき、メモリに比べて処理速度の遅い物理ディスク装置への入出力回数を削減するためのメモリ領域）やファイル管理テーブル、及び、ディスク装置上のディスクキャッシュ（バッファキャッシュと同様の目的のために物理ディスク装置が備えるキャッシュメモリ）の内容が正常に物理ディスク装置に書き戻されず、大切なデータが消えてしまうという問題点をも生じる。しかも、これが原因でファイルシステムの整合性が異常となるため、物理ディスク装置

に冗長に記録されたファイルシステムの情報を元にファイルシステムの整合性を正常状態に戻す操作が必要となる。この操作は、ディスク装置全体をチェックする必要があるため、長い時間を要する。この結果、この間、その物理ディスク装置に対するＩＯ処理を行うことはできないという問題点を生じさせてしまう。

【００１０】さらに、ＩＯパス切り替え後、新しいＩＯパスを用いてディスク装置にアクセスを行うので、ＩＯパス切り替え後にシステムを再起動したときにマウント処理が正常に行われるようにするには、システム管理者がマウント構成ファイルを更新し、ディスク装置への新しいＩＯパスとマウントポイントとの対応関係をマウント構成ファイルに登録しなおす必要がある。また、ファイルのミラーリングを行う場合、論理ボリュームを作成する必要があるが、論理ボリュームの管理は、システム管理者に対して煩雑な作業を行わせることになる。

【００１１】本発明の第１の目的は、ＩＯパスの切り替え処理のために要する時間を短縮し、一般ユーザからＩＯパス切り替え処理をできるだけ隠蔽することができるファイルシステムを提供することにある。また、本発明の第２の目的は、ＩＯパスの切り替え時に、バッファキャッシュやファイル管理テーブル及びディスク装置上のディスクキャッシュに保存されたデータを失うことなくＩＯパスの切り替え処理を行い、ファイルの整合性のチェックを不要とすることができるファイルシステムを提供することにある。また、本発明の第３の目的は、ＩＯパスを切り替えたとき自動的にマウント構成ファイルを更新し、システム管理者の負担を軽減することのできるファイルシステムを提供することある。さらに、本発明の第４の目的は、ユーザに論理ボリュームを意識させずにファイルのミラーリングを行う方法を備えたファイルシステムを提供することにある。

【００１２】

【課題を解決するための手段】本発明によれば前記目的は、ファイル毎にファイルＩＤが定義されており、複数の物理ディスク装置に分散管理されたファイルの処理を行う１または複数のファイルサーバを有するファイルシステムにおいて、ファイルＩＤ及び該ファイルＩＤに対応するファイルが格納されている論理ディスクの論理ディスクＩＤを含むファイル管理テーブルと、論理ディスクＩＤ及び前記論理ディスクに対応する１つ以上の物理ディスク装置にアクセスするための１つ以上のＩＯパスを含む論理ディスク管理テーブルとを備え、ユーザからのファイルＩＤを指定したファイルへのアクセス要求を受信したファイルサーバは、ファイル管理テーブルを参照し、前記ファイルＩＤから前記ファイルが格納されている論理ディスクの論理ディスクＩＤを決定し、論理ディスク管理テーブルを参照して前記論理ディスクＩＤから前記論理ディスクに対応する物理ディスク装置にアクセスするためのＩＯパス（ＩＯパスを決定する情報は、ノード番号、ＩＯインターフェイス番号、ディスクコントローラ番号である）を決定し、決定したＩＯパスを使用して物理ディスク装置にアクセスすることにより達成される。

【００１３】前述において、論理ディスク管理テーブルは、該論理ディスク管理テーブルに登録されているＩＯパス毎に稼働状態（「使用中」、「待機中」、「使用不可」）を保持する状態フラグを含み、通常運用時、ファイルサーバは状態フラグが「使用中」状態のＩＯパス（運用系ＩＯパス）を用いて物理ディスク装置にアクセスする。前記運用系ＩＯパスの障害発生時、障害を検出したノードのファイルサーバは、前記ノードの論理ディスク管理テーブルを更新し、前記障害発生ＩＯパスの状態フラグを「使用不可」とし、状態フラグが「待機中」状態であるＩＯパスの状態フラグを「使用中」として新運用系ＩＯパスとした後、全リモートノードのファイルサーバと通信を行い、前記論理ディスク管理テーブルの内容を全ノードの論理ディスク管理テーブルに複写することによって、前記物理ディスク装置にアクセスするためのＩＯパスを旧運用系ＩＯパスから新運用系ＩＯパスに切り替える。

【００１４】このＩＯパス切り替え処理の間、前記障害発生ＩＯパスに含まれるノードのファイルサーバは、旧運用系ＩＯパスへのアクセス要求を保留し、ＩＯパス切り替え処理終了時、保留していたアクセス要求を前記新運用系ＩＯパスが含むノードに送信する。これによって、ＩＯパス切り替え処理を動的に行うことが可能となり、ＩＯパス切り替え時ファイル管理テーブルを検索・更新する必要をなくし、ＩＯパス切り替え処理に要する時間を短縮することができる。

【００１５】また、前述において、ＩＯパスの切り替え処理時、使用できなくなった旧運用系ＩＯパスを使ってアクセスしていた物理ディスク装置内に設けられたディスクコントローラが有するディスクキャッシュに格納されたデータのうち、前記物理ディスク装置に書き戻す必要のあるデータを、前記物理ディスク装置内に設けられた別のディスクコントローラを使用して前記物理ディスク装置に書き戻し、前記旧運用系ＩＯパスに含まれるノードのファイルサーバと新運用系ＩＯパスに含まれるノードのファイルサーバが通信を行うことによって、前記旧運用系ＩＯパスに含まれるノードの主記憶内に存在し、前記物理ディスク装置に書き戻す必要があるバッファキャッシュ及びファイル管理テーブルを前記新運用系ＩＯパスに含まれるノードに転送する。本発明は、これによって、ディスク装置上のディスクキャッシュに存在していたデータや、バッファキャッシュや、ファイル管理テーブルが消失するのを防ぎ、ファイルシステムの整合性のチェックを不要とすることができる。

【００１６】また、前述において、マウント構成ファイルは、ＩＯパス毎にそのＩＯパスが使用できるか否かを

登録する使用可否情報を含み、ファイルサーバは、システム起動時に前記マウント構成ファイルを読み込み、前記使用可否情報に「使用可」と記載されたＩＯパスについて、対応する論理ディスク管理テーブルの状態フラグを「使用中」または「待機中」と登録し、前記使用可否情報に「使用不可」と記載されたＩＯパスについて、対応する論理ディスク管理テーブルの状態フラグを「使用不可」と登録することにより、マウント構成ファイルに「使用可」と記載されたＩＯパスだけを使用して物理ディスク装置にアクセスをする設定を行っている。ＩＯパス切り替え・切り離し処理終了後、ファイルサーバは、前記マウント構成ファイルを更新し、使用できなくなった旧運用系ＩＯパスの使用可否情報を「使用不可」に書き換える。また、使用不可能となったＩＯパスが再び使用できるようになったとき、ファイルサーバは、マウント構成ファイルを更新し、使用可能になった前記ＩＯパスの使用可否情報を「使用可」に書き換える。このように、本発明は、ＩＯパスが切り替わったときや復旧したときのマウント構成ファイルの書き換え処理を自動化することにより、システム管理者の負担を軽減することができる。

【００１７】また、本発明は、マウント構成ファイルの１つのエントリに書かれた複数のＩＯパスからアクセスされる複数のディスク装置に対して、ファイルのミラーリングを行うことができ、これにより、ユーザが論理ボリュームを使用することなくファイルのミラーリングを行うことができる。

【００１８】

【発明の実施の形態】以下、本発明によるファイルシステムの実施形態を図面により詳細に説明する。

【００１９】図１は本発明の第１の実施形態によるファイルシステムの構成を示すブロック図、図２はシステム内に設けられる各種のテーブルの具対的な構成例を説明する図、図３はマウント構成ファイルの具体的な構成例を説明する図である。図１～図３において、１はネットワーク、１０、２０、３０は物理ディスク装置、１１、１２、２１、２２はディスクコントローラ、２４はマウント構成ファイル、１００、２００、３００はノード、１１０、２１０、３１０はＣＰＵ、１２０、２２０、３２０はメモリ、１３０、２３０はユーザアプリケーション（ＵＡＰ）、１４０、２４０はファイルサーバ（ＦＳ）、２５０はディスクドライバ、１６０、２６０はファイル管理テーブル、１７０、２７０は論理ディスク管理テーブル、１８０、２８０はバッファキャッシュ、２９０、３９０はＩＯインタフェースである。

【００２０】本発明の第１の実施形態によるファイルシステムは、図１に示すように、超並列計算機システムを構成するノード１００、２００、３００（図１では３つのノードのみを示しているが、ノードは多数設けられる）がネットワーク１によって相互に接続されて構成されている。ノード２００とノード３００とには、両ノードからアクセス可能な共用物理ディスク装置１０、２０が接続されている。物理ディスク装置１０、２０は、それらのディスク装置内に設けられたディスクコントローラ１１、１２及びノード２００内に設けられたＩＯインターフェイス２９０によってノード２００と接続されると共に、ディスクコントローラ１２、２２及びノード３００内に設けられたＩＯインターフェイス３９０によってノード３００と接続されている。ノード１００に接続されている物理ディスク装置３０は、物理ディスク装置１０、２０と比べて障害発生率が極めて低い高信頼ディスク装置である。

【００２１】ノード２００は、ＣＰＵ２１０とメモリ２２０とから構成される。メモリ２２０は、ユーザアプリケーション２３０と、ファイル制御を行うファイルサーバ２４０と、ディスクＩＯ処理を行うディスクドライバ２５０と、ファイル管理テーブル２６０と、論理ディスクを定義している論理ディスク管理テーブル２７０と、バッファキャッシュ２８０とを含む。ノード１００及びノード３００は、ノード２００と同様に構成されている。

【００２２】物理ディスク装置にアクセスするための入出力経路をＩＯパスと呼び、このＩＯパスは、ノード番号、ＩＯインターフェイス番号、ディスクコントローラ番号の３つの情報で決定され、ＩＯパスを決めると物理ディスク装置を一意に決めることができる。例えば、（ノード番号、ＩＯインターフェイス番号、コントローラ番号）＝（２００，２９０，１１）というＩＯパスからは、物理ディスク装置１０にアクセスされる。以後の説明において、ＩＯパスは、前述のような形式で記載することとする。

【００２３】論理ディスクは、１つ以上の物理ディスク装置を組み合わせたものとして構成される。その物理ディスクの組み合わせは、ＩＯパスを指定することによって行われる。例えば、（２００，２９０，１１）、（３００，３９０，２２）という２つのＩＯパスを組み合わせると、物理ディスク装置１０、２０を纏めた論理ディスクを構成することができる。その際、物理ディスク装置１０、２０に同一の内容を記録するようにすれば、論理ディスクをミラー化することができる。また、（２００，２９０，１１）、（３００，３９０，１２）という２つのＩＯパスを組み合わせると、これらのＩＯパスからは共に物理ディスク装置１０にアクセスされるため、物理ディスク装置１０に対応する論理ディスクが構成される。但し、この場合、物理ディスク装置１０にアクセスするためのＩＯパスが２通り存在するので、片方のＩＯパスに障害が発生した場合でも、別のＩＯパスから物理ディスク装置１０にアクセスすることができ、これによって、ディスク装置の信頼性の向上を図ることができる。説明する本発明の第１の実施形態は、論理ディスク

が１つの物理ディスク装置に対応する後者の場合を例として取り扱う。

【００２４】論理ディスク管理テーブル２７０は、図２（ｂ）に示すように、論理ディスクＩＤ２７１と、ノード番号２７２、２７６と、ＩＯインターフェイス番号２７３、２７７と、ディスクコントローラ番号２７４、２７８と、状態フラグ２７５、２７９とから構成される。２７２～２７４は、論理ディスクＩＤ２７１に対応する物理ディスク装置にアクセスするための第１のＩＯパスを決定し、状態フラグ２７５には、このＩＯパスの稼働状態（「使用中」、「待機中」、「使用不可」のいずれか）が登録される。２７６～２７８は、物理ディスク装置にアクセスするための第２のＩＯパスを決定し、このＩＯパスの稼働状態が状態フラグ２７９に登録される。このように論理ディスク管理テーブル２７０には、１つの論理ディスクＩＤに対して２通りのＩＯパスとそれぞれのＩＯパスの状態フラグを登録できるようになっている。

【００２５】本発明の第１の実施形態において、前述の２つのＩＯパスからアクセスされる物理ディスク装置は同一のものであり、通常運用時は２つのＩＯパスのうち１つを使用し（状態フラグが「使用中」状態になっている）、もう一方のＩＯパスを「待機中」状態としておき、ディスクコントローラやＩＯインターフェイスの障害等の原因により、使用中のＩＯパスが使用できなくなったとき、ファイルサーバが物理ディスク装置にアクセスするためのＩＯパスを「待機中」状態のＩＯパスに切り替える。このように、論理ディスク管理テーブルは、論理ディスクＩＤと物理ディスク装置にアクセスするためのＩＯパスとを対応付けることによって、仮想的なディスク装置として論理ディスクを定義している。論理ディスクＩＤはこの論理ディスクを識別するための番号である。

【００２６】また、システムを構成する各ノードが持つ論理ディスク管理テーブルの内容は常に同一となっている。例えば、図１において、ノード１００が持つ論理ディスク管理テーブル１７０と、ノード２００が持つ論理ディスク管理テーブル２７０と、ノード３００が持つ論理ディスク管理テーブル３７０は常に同一の内容を有する。

【００２７】ファイル管理テーブル２６０は、図２（ａ）に示すように、ファイルＩＤ２６１と論理ディスクＩＤ２６２とファイル管理情報２６３とにより構成される。ファイルＩＤ２６１には、現在オープンされているファイルのファイルＩＤが登録され、論理ディスクＩＤ２６２には、前述のファイルが格納されている論理ディスクの論理ディスクＩＤが登録される。ファイル管理情報２６３には、前述のファイルのファイルサイズや更新日付等の情報が登録される。このファイル管理テーブル２６０の各エントリは、ノード２００上で動作するプログラムがファイルをオープンする度に、物理ディスク装置上から各ファイル固有の情報として読み出される。従って、ファイル管理テーブル２６０のエントリは、少なくともオープンされているファイルの個数分存在する。

【００２８】バッファキャッシュ２８０は、物理ディスク装置にアクセスを行うときにリード・ライトするデータを一時的に蓄えておき、メモリに比べて処理速度の遅い物理ディスク装置への入出力処理回数を削減するために使用される。バッファキャッシュ２８０は、図２（ｃ）に示すように、論理ディスクＩＤ２８１とブロック番号２８２とキャッシュデータ２８３とから構成される。キャッシュデータ２８３には、論理ディスクＩＤ２８１のブロック番号２８２で指定されるディスク領域のデータの内容が格納される。

【００２９】高信頼な物理ディスク装置３０内には、マウント構成ファイル２４が格納されている。マウント構成ファイル２４のエントリは、図３に示すように、システムに接続される物理ディスク装置にアクセスするためのＩＯパス名５１、５３と、そのＩＯパスが使用可能か否かを示す使用可否情報５２、５４と、前述の物理ディスク装置に対応する論理ディスクをマウントするマウントポイント５５との３つの情報を含んでいる。マウント構成ファイル２４には、ＩＯパス名が“（ノード番号，ＩＯインターフェイス番号，ディスクコントローラ番号）＝（２００，２９０，１１）”のような形式で記述され、そのＩＯパスが使用可能な場合、マウント構成ファイル２４の対応するＩＯパスの使用可否情報に“available”と記述され、そのＩＯパスが使用不可能な場合、使用可否情報に“unavailable”と記述される。図３に示した例では、ＩＯパス（２００，２９０，１１）と（３００，３９０，１２）との両者がマウントポイント ／mntに対応付けされており、共に使用可能となっている。この記述によって、ユーザが ／mntディレクトリ以下のディレクトリツリー内のファイルにアクセスしたとき、物理ディスク装置１０にアクセスできるようになる。このとき、物理ディスク装置１０にアクセスするためのＩＯパスは、前述のいずれかのＩＯパスが使用される。使用していない方のＩＯパスは「待機中」状態としてスタンバイしている。

【００３０】前述のように、物理ディスク装置にアクセスするためのＩＯパスが２つ存在する場合、その２つのＩＯパスを同じエントリに記載することにより、２つのＩＯパスを１つのマウントポイントに対応付けることができる。マウント構成ファイル２４は、通常のエディタなどで編集することが可能であり、システム管理者は、システムの構成を変更したとき、マウント構成ファイル２４の内容が新しいシステム構成と一致するように、マウント構成ファイル２４を編集し、システムをリブートさせる。システムの起動時、ファイルサーバ１４０は、

修正後のマウント構成ファイル２４に従ってマウント処理を行うので、リブート後、新しいシステム構成が使用可能となる。例えば、図１に示した物理ディスク装置装置２０をシステムに追加したとき、システム管理者は“（（２００，２９０，２１）available）（（３００，３９０，２２）available）／mnt1”という行をマウント構成ファイル２４に追加してシステムをリブートする。この記述によって、ユーザが／mnt1 ディレクトリにアクセスしたとき、前述の追加行に記載したいずれかのＩＯパスから物理ディスク装置２０にアクセスできるようになる。

【００３１】図４はシステムの起動時のファイルサーバの処理動作を説明するフローチャート、図５はシステム全体のノードの論理ディスク管理テーブルを更新する処理動作を説明するフローチャートであり、次に、これらのフローを参照して、システムの起動時にファイルサーバ１４０がマウント構成ファイル２４を読み込み、論理ディスク管理テーブルを設定してマウント処理を行うまでの処理手順及び全ノードでの論理ディスク管理テーブルの更新の処理手順を説明する。

【００３２】（１）システムの起動時、ノード１００内のファイルサーバ１４０は、高信頼ディスク装置３０上に格納されているマウント構成ファイル２４の１つのエントリを読み込む（ステップ４０１、４０２）。

【００３３】（２）ファイルサーバ１４０は、マウント構成ファイル２４に記載されたＩＯパス名に対して論理ディスクＩＤを自動的に設定する。マウント構成ファイル２４の１つのエントリに複数のＩＯパス名が記載されていた場合、ファイルサーバ１４０は、その複数のＩＯパスに対して１つの論理ディスクＩＤを設定する。例えば、図３に示した例の場合、ファイルサーバ１４０は、ＩＯパス名５１“（２００，２９０，１１）”及びＩＯパス名５３“（３００，３９０，１２）”に対して論理ディスクＩＤ“１２３”を設定する。ファイルサーバ１４０は、これにより、設定した論理ディスクＩＤを論理ディスク管理テーブル１７０の論理ディスクＩＤ１７１に登録する（ステップ４０３）。

【００３４】（３）前述の第１のＩＯパス名をノード番号１７２、ＩＯインターフェイス番号１７３、ディスクコントローラ番号１７４に登録し、第２のＩＯパス名をノード番号１７６、ＩＯインターフェイス番号１７７、ディスクコントローラ番号１７８に登録する。図３に示した例の場合、論理ディスクＩＤ１７１には“１２３”、ノード番号１７２には“２００”、ＩＯインターフェイス番号１７３には“２９０”、ディスクコントローラ番号１７４には“１１”、ノード番号１７６には“３００”、ＩＯインターフェイス番号１７７には“３９０”、ディスクコントローラ番号１７８には“１２”が登録される（ステップ４０４）。

【００３５】（４）そして、ファイルサーバ１４０は、マウント構成ファイル２４の使用可否情報に“available”と記載されている最初のＩＯパス“（２００，３９０，１１）”について、論理ディスク管理テーブル１７０の対応する状態フラグを「使用中」状態と登録し、“available”と記載されている残りのＩＯパス“（３００，３９０，１２）”について、対応する状態フラグを「待機中」状態と登録する。また、ファイルサーバ１４０は、マウント構成ファイル２４の使用可否情報に、“unavailable”と記載されているＩＯパスについては対応する状態フラグを「使用不可」状態と登録する。この結果、論理ディスク管理テーブル１７０の内容は、図２に示したようなものとなる（ステップ４０５）。

【００３６】（５）ファイルサーバ１４０は、マウント構成ファイル２４に記載された全てのエントリについて、論理ディスク管理テーブル１７０への登録が終了したか否かをチェックし、終了していない場合、ステップ４０２からの処理を繰り返し実行して論理ディスク管理テーブルへの登録を続ける（ステップ４０６）。

【００３７】（６）ステップ４０６で、マウント構成ファイル２４に記載された全てのエントリについて、論理ディスク管理テーブル１７０への登録が終了していた場合、ファイルサーバ１４０は、全ての他のノード２００、３００であるリモートノードのファイルサーバと通信を行い、システムを構成する全ノードの論理ディスク管理テーブルの更新を行わせる（ステップ４０７）。

【００３８】（７）ファイルサーバ１４０は、全リモートノードから論理ディスク管理テーブルの更新完了の通知を受信したら、マウント構成ファイル２４に記載されているＩＯパス名（“（２００，２９０，１１）”及び“（３００，３９０，１２）”）とマウントポイント／mnt との対応関係、及び、論理ディスク管理テーブル１７０に登録した上記ＩＯパス名と論理ディスクＩＤ“１２３”との対応関係から、マウントポイント /mnt と上記論理ディスクＩＤ“１２３”との対応関係を作り、論理ディスクＩＤ“１２３”に対応する論理ディスクをマウントポイント /mnt にマウントする（ステップ４０８）。

【００３９】次に、図５に示すフローを参照して前述したステップ４０７の処理時のファイルサーバ１４０及びリモートノードのファイルサーバの処理動作を説明する。

【００４０】（１）ファイルサーバ１４０は、自ノード１００の論理ディスク管理テーブルの設定を終了した後、全リモートノードのファイルサーバに論理ディスク管理テーブル１４０の内容を送信し、論理ディスク管理テーブルを更新するように要求する（ステップ９０１、９０２）。

【００４１】（２）この通知を受けたリモートノードのファイルサーバは、送信されてきた論理ディスク管理テーブル１７０の内容を、そのノードの論理ディスク管理

テーブルに複写して論理ディスク管理テーブルの更新を行い、ファイルサーバ１４０に論理ディスク管理テーブルの更新終了を通知する（ステップ９０５～９０７）。

【００４２】（３）ファイルサーバ１４０は、全リモートノードからそれぞれのノードの論理ディスク管理テーブルの更新完了通知を受信するのを待ち、図４により説明したステップ４０８のマウント処理を実行して処理を終了する（ステップ９０３、９０４）。

【００４３】図６は通常運用時のファイルサーバの処理動作を説明するフローチャートであり、次に、このフローを参照して、通常運用時のファイルアクセスの手順について説明する。ここでは、ファイル管理テーブル１６０、２６０及び論理ディスク管理テーブル１７０、２７０の設定が図２に示すようになっているとして、ローカルノードとしてのノード２００に接続された物理ディスク装置にアクセスする場合について、ノード２００上で動作するユーザアプリケーション２３０が、ファイルＩＤ“１００”を指定したファイルアクセス要求をファイルサーバ２４０に発行した場合を例に説明する。

【００４４】（１）ファイルサーバ２４０は、ユーザアプリケーション２３０からの要求を受信すると、この要求が他のノードであるリモートノードからの要求であるか否かを判定する（ステップ５０１、５０２）。

【００４５】（２）説明している例では、自ノードであるローカルノードのユーザアプリケーションからのアクセスであるとしているので、ファイルサーバ２４０は、ファイル管理テーブル２６０を検索し、ファイルＩＤ“１００”からそのファイルＩＤで定義されるファイルが格納されている論理ディスクの論理ディスクＩＤ“１２３”を求める（ステップ５０３）。

【００４６】（３）そして、ファイルサーバ２４０は、論理ディスク管理テーブル２７０を検索し、論理ディスクＩＤから状態フラグが「使用中」状態のＩＯパス名“（２００，２９０，１１）”を求め、そのＩＯパス名に含まれるノード番号“２００”がローカルノードであるか否かを判定する（ステップ５０４、５０５）。

【００４７】（４）前述のＩＯパス名に含まれるノード番号“２００”がローカルノードであるとして説明しているので、ステップ５０５で、前述のＩＯパス名に含まれるノード番号“２００”がローカルノードであると判定され、ファイルサーバ２４０は、自ローカルノードのディスクドライバ２５０にＩＯパスを指定したＩＯアクセス要求を送る。この要求を受けたディスクドライバ２５０は、ＩＯインターフェイス２９０を介してディスクコントローラ１１に制御信号を送る（ステップ５０７）。

【００４８】次に、他のノードであるリモートノードに接続された物理ディスク装置にアクセスする場合について説明する。ここで説明する例は、ノード１００上で動作するユーザアプリケーション１３０が、ファイルＩＤ“１００”を指定したファイルアクセス要求をファイルサーバ１４０に発行した場合であるとする。

【００４９】（１）ファイルサーバ１４０は、ユーザアプリケーション１３０からの要求を受信すると、ローカルノードに接続された物理ディスク装置にアクセスする場合と同様に、ファイル管理テーブルを１６０を検索しファイルＩＤ“１００”から論理ディスクＩＤ“１２３”を求め、論理ディスク管理テーブル１７０を検索して論理ディスクＩＤ“１２３”からＩＯパス名“（２００，２９０，１１）”を求める（ステップ５０１～５０４）。

【００５０】（２）ファイルサーバ１４０は、上記ＩＯパス名に含まれるノード番号“２００”がリモートノードであることを確認すると、そのノード（ノード２００）のファイルサーバ２４０に上記論理ディスクＩＤを指定したＩＯアクセス要求を送る（ステップ５０５、５０６）。

【００５１】（３）この要求を受けたファイルサーバ２４０は、論理ディスク管理テーブル２７０を検索し論理ディスクＩＤ“１２３”から状態フラグが「使用中」状態のＩＯパス名“（２００，２９０，１１）”を求める（ステップ５０１、５０２、５０４）。

【００５２】（４）ファイルサーバ２４０は、ＩＯパスに含まれるノード番号“２００”が自ノードであるローカルノードであることを確認して、ディスクドライバ２５０にＩＯパスを指定したＩＯアクセス要求を送る。この要求を受けたディスクドライバ２５０は、ＩＯインターフェイス２９０を介してディスクコントローラ１１に制御信号を送る（ステップ５０５、５０７）。

【００５３】前述した処理動作の説明から判るように、ファイルサーバが自ノードであるローカルノードからアクセス要求を受ける場合、その要求は、全てユーザアプリケーションからの要求であり、他のノードであるリモートノードからの要求を受ける場合、その要求は、全てリモートノードのファイルサーバからの要求である。

【００５４】実際のファイルアクセス処理は、バッファキャッシュを経由して行われる。ファイルサーバ２４０は、論理ディスクＩＤを指定したＩＯアクセス要求に対する処理を、バッファキャッシュ２８０に対するリード・ライト処理と、バッファキャッシュ２８０と物理ディスク装置１０との間でのリード・ライト処理とに分けて行う。ファイルサーバ２４０は、バッファキャッシュ２８０と物理ディスク装置１０との間のリード・ライトアクセス処理との実行時に、論理ディスクＩＤからＩＯパス名への変換を行う。ノード１００で動作するプログラムが、リモートノードに接続された物理ディスク装置１０にアクセスする場合、ノード１００上のバッファキャッシュ１８０とノード２００上のバッファキャッシュ２８０とを経由してアクセスが行われる。すなわち、ライト処理を行う場合のデータの流れは、バッファキャッシ

ュ１８０→バッファキャッシュ２８０→物理ディスク装置１０となる。リード処理の場合、この逆の順序となる。

【００５５】ユーザアプリケーションがファイルを更新し、ファイルの更新日付が変わるなどして、ファイル管理テーブルの内容が変更されたとき、ファイル管理テーブルの変更を物理ディスク装置に書き戻す必要がある。次に、この書き戻し処理について説明する。

【００５６】ファイル管理テーブルの内容が変更され、その内容をローカルノードに接続された物理ディスク装置に書き戻す場合、ローカルノードのファイルサーバがローカルノードのファイル管理テーブルの内容を直接その物理ディスク装置に書き戻す。また、リモートノードに接続された物理ディスク装置に書き戻す場合、ローカルノードのファイルサーバは、物理ディスク装置が接続されたノードにローカルノードのファイル管理テーブルの内容を一旦転送する。その後、物理ディスク装置が接続されたノードのファイルサーバが物理ディスク装置にその内容を書き戻す。例えば、ノード１００のファイルサーバ１４０がファイル管理テーブル１６０の内容を物理ディスク装置１０に書き戻す場合、まず、ファイルサーバ１４０は、物理ディスク装置への書き戻し処理を行いたいファイル管理テーブル１６０のエントリ中の、論理ディスクＩＤ１６２（“１２３”）を参照して、書き戻す先の論理ディスクＩＤを求める。そして、論理ディスク管理テーブル１７０を検索して上記論理ディスクＩＤに対応する物理ディスク装置にアクセスするためのＩＯパス（“２００，２９０，１１”）を求め、そのＩＯパス名に含まれるノード番号（“２００”）に対応するノード（ノード２００）のファイルサーバ２４０に書き戻しを行いたいファイル管理テーブルのエントリを送信する。ファイルサーバ２４０は、受信したデータを一旦ファイル管理テーブル２６０に書き込む。その後、ファイルサーバ２４０は、ファイル管理テーブルに保存されている他のデータと纏めて、ファイル管理テーブル２６０の更新内容を物理ディスク装置１０に書き込む。ファイルサーバ２４０が物理ディスク装置１０にアクセスするためのＩＯパスは、論理ディスク管理テーブル２７０を検索し、論理ディスクＩＤ２６２をＩＯパス名に変換することによって求められる。

【００５７】前述したように、最終的な物理ディスク装置へのデータの書き戻しは、物理ディスク装置が接続されたノードに存在するファイル管理テーブル及びバッファキャッシュから行っており、物理ディスク装置が接続されたノードのファイル管理テーブル及びバッファキャッシュには、ローカルノードのユーザアプリケーションに関係するもの以外にリモートノードのユーザアプリケーションに関係するものが存在する。

【００５８】図７はＩＯパスの切り替えの処理動作を説明するフローチャート、図８～図１０はＩＯパスに障害が発生しＩＯパスの切り替えを行う処理について説明する図である。図８～図１０において、１３はディスクキャッシュ、３４０はファイルサーバ、３５０はディスクドライバ、３６０はバッファキャッシュであり、他の符号は図１の場合と同一である。以下、これらの図を参照して、ディスクコントローラ１１で障害が発生し、通常使用しているＩＯパス“（２００，２９０，１１）”が使用不可能になったとき、物理ディスク装置１０にアクセスするためのＩＯパスを“（２００，２９０，１１）”から“（３００，３９０，１２）”に切り替える処理について説明する。

【００５９】図９において、ディスクキャッシュ１３は、ディスク装置１０が備えるディスクコントローラ１１の内部に設けられたディスクキャッシュであり、ディスクコントローラ１１に対してリード・ライト処理要求が発行されたときに使用される。そして、実際のリード・ライト処理は、このディスクキャッシュ１３を経由して行われる。また、ディスクコントローラ１２は、ディスクコントローラ１１に障害が発生したときに、ディスクキャッシュ１３がディスク媒体に書き戻す必要のあるデータを保持している場合、そのデータをディスク媒体に書き戻し、ディスクコントローラ１１をディスク装置から切り放す機能を持つ。

【００６０】図８は図７により説明するステップ１００３でのリクエストの保留の処理を行うときの各ノードの動作を示し、図９は図７により説明するステップ１００４でのディスクキャッシュの書き戻しの処理と、ステップ１００５でのバッファキャッシュの転送の処理を行うときの各ノードの動作を示し、図１０は図７により説明するステップ１００６でのリクエストの保留解除及び転送の処理を行うときの各ノードの動作を示している。

【００６１】以下、ディスクコントローラ１１で障害が発生した時、物理ディスク装置１０にアクセスするためのＩＯパスを“（２００，２９０，１１）”から“（３００，３９０，１２）”に切り替える処理を図８～図１０を併用しながら図７に示すフローを参照して説明する。なお、論理ディスク管理テーブル２７０の設定は図２に示すようになっているものとする。

【００６２】障害検出の処理（ステップ１００１）
ディスクコントローラ１１に障害が発生すると、ディスクドライバ２５０は、ＩＯパス（２００，２９０，１１）を使って物理ディスク装置１０にアクセスを行うことができなくなる。これをもって障害検出とし、ディスクドライバ２５０は、ＩＯパス（２００，２９０，１１）の障害発生をファイルサーバ２４０に通知する。また、ディスクドライバ２５０がローカルノードとしてのノード２００のノード番号を含むＩＯパスのうち、論理ディスク管理テーブル２７０の状態フラグが、「使用中」状態及び「待機中」状態のＩＯパスを定期的に監視することによって障害を検出してもよい。これによっ

て、「待機中」状態のＩＯパスの障害検出が可能となる。

【００６３】切り替え対象ＩＯパスの検索の処理（ステップ１００２）

障害発生通知を受けたファイルサーバ２４０は、図２に示した論理ディスク管理テーブル２７０を参照し、障害発生ＩＯパス“（２００，２９０，１１）”を含むエントリを検索する。そして、障害発生ＩＯパスの状態フラグが「待機中」状態であるか否かをチェックし（ステップ１０１０）、もし、障害発生ＩＯパスの状態フラグが「待機中」状態であれば、ＩＯパスの切り替え処理は必要なく、ステップ１０１１の処理に進む。そうでない場合、ＩＯパスの切り替えが必要になりステップ１１０３の処理に進む。前述の検索によって見つかったエントリには、障害発生ＩＯパス以外に、状態フラグ２７９（２７５）が「待機中」状態のＩＯパス“（３００，３９０，１２）”と論理ディスクＩＤ“１２３”が登録されている。この「待機中」状態のＩＯパス“（３００，３９０，１２）”が切り替え先のＩＯパスとなる。ファイルサーバ２４０は、障害発生ＩＯパス名と切り替え先のＩＯパス名とそれらに対応する論理ディスクＩＤ（以後、ＩＯパス切り替え処理を行う論理ディスクＩＤと呼ぶ）を、ファイルサーバ２４０が管理するメモリ内に保存し、ファイルサーバ２４０が論理ディスク管理テーブル２７０を検索することなくいつでも得られるようにしておく。

【００６４】リクエストの保留の処理（ステップ１００３）

この処理について、図８を参照して説明する。ファイルサーバ２４０は、現在処理中あるいは今後受理するＩＯアクセス要求の中で、ＩＯパスの切り替え処理を行う論理ディスクＩＤ“１２３”あるいは障害発生ＩＯパス“（２００，２９０，１１）”を指定したＩＯアクセス要求を保留し、その内容を後で取り出すことができるようにファイルサーバ２４０が管理するメモリ上に記録する。図８に示す例では、ファイルサーバ１４０は、ディスクコントローラ１１で障害が発生したことを知らずに、論理ディスク“１２３”を指定したライト要求をファイルサーバ２４０に送信している。ファイルサーバ２４０は、このライト要求と、現在処理中のＩＯパス名“（２００，２９０，１１）”を指定したリード要求を保留している。

【００６５】次に、ファイルサーバ２４０は、切り替え先のＩＯパス“（３００，３９０，１２）”に含まれるノード番号“３００”に対応するノード（以後、切り替え先のノードと呼ぶ）のファイルサーバ３４０に、障害発生ＩＯパス名“（２００，２９０，１１）”と切り替え先のＩＯパス名“（３００，３９０，１２）”と対応する論理ディスクＩＤ“１２３”とを送信し、論理ディスクＩＤを指定したＩＯアクセス要求を保留するように要求する。この要求を受信したファイルサーバ３４０は、前述の２つのＩＯパス名と論理ディスクＩＤとをファイルサーバ３４０が管理するメモリ上に保存し、これらの情報をいつでも得られるようにした後、論理ディスクＩＤ“１２３”を指定したＩＯアクセス要求を保留し、その内容を後で取り出せるようにファイルサーバ３４０が管理するメモリ上に保存する。図８に示す例では、ファイルサーバ３４０は、論理ディスクＩＤ“１２３”を指定したリード要求を保留している。

【００６６】ディスクキャッシュの書き戻しの処理（ステップ１００４）

この処理について、図９を参照して説明する。ファイルサーバ３４０は、リクエストの保留の設定を行った後、障害発生ＩＯパスが含むディスクコントローラ番号“１１”に対応するディスクコントローラ１１が備えるディスクキャッシュ１３を、切り替え先のＩＯパスが含むディスクコントローラ番号“１２”に対応するディスクコントローラ１２を使ってディスク装置に書き戻すようにディスクドライバ３５０に要求する。この要求を受けたディスクドライバ３５０は、ＩＯインターフェイス３９０を介してディスクコントローラ１２に制御信号を送りディスクキャッシュ１３に保存されているｄｉｒｔｙなデータをディスク領域に書き戻し、ディスクコントローラ１１をディスク装置１０から切り放す。これらの処理の終了後、ディスクドライバ３５０は、ファイルサーバ３４０に終了通知を送る。

【００６７】バッファキャッシュの転送の処理（ステップ１００５）

この処理について、図９を用いて説明する。ファイルサーバ３４０は、ディスクドライバ３５０からの終了通知を受けると、障害発生ＩＯパス“（２００，２９０，１１）”に含まれるノード番号“２００”に対応するノード（以後、障害発生ノードと呼ぶ）のファイルサーバ２４０にファイル管理テーブル２６０及びバッファキャッシュ２８０の転送を要求する。ファイルサーバ３４０からの要求を受信したファイルサーバ２４０は、ｄｉｒｔｙな（物理ディスク装置に書き戻す必要のある）ファイル管理テーブル２６０とｄｉｒｔｙなバッファキャッシュ２８０の中で、論理ディスクＩＤ２６２や論理ディスクＩＤ２８１が、ＩＯパス切り替え処理を行う論理ディスクＩＤ“１２３”であるデータを、ファイルサーバ３４０に送信する。この送信が成功したら、ファイルサーバ２４０は、ノード２００内に存在する前述のデータを消去可能とし、バッファキャッシュ２８０をしばらくの間、読み出し用のキャッシュとして使用するが、バッファキャッシュ２８０やファイル管理テーブル２６０のためのメモリ領域が不足してきたらこれらを消去する。ファイルサーバ３４０は、受け取ったデータを、ノード３００上のファイル管理テーブル３６０及びバッファキャッシュ３８０にマージする。ノード３００上のこれらの

データはｄｉｒｔｙであるので、ＩＯパスの切り替え処理が終了し通常運用状態となったら、ファイルサーバ３４０が切り替え先のＩＯパス“（３００，３９０，１２）”を使用して物理ディスク装置１０に書き込む。また、前述の上記データは、読み出し用のキャッシュとして使用される可能性もある。

【００６８】論理ディスク管理テーブルの更新の処理（ステップ１００６）

この処理は、図５により説明したフローの手順で実行される。図５に示したローカルノードは、ここでは障害発生ノード２００である。ファイル管理テーブル２６０及びバッファキャッシュ２８０の転送が終了すると、ファイルサーバ２４０は、論理ディスク管理テーブル２７０に登録されている障害発生ＩＯパス“（２００，２９０，１１）”の状態フラグ２７５を「使用中」状態から「使用不可」状態に、切り替え先のＩＯパス“（３００，３９０，１２）”の状態フラグ２７９を「待機中」状態から「使用中」状態に更新する。ファイルサーバ２４０は、論理ディスク管理テーブル２７０の更新の終了後（図５のステップ９０１）、全リモートノードのファイルサーバに論理ディスク管理テーブル２７０の更新情報を送り、論理ディスク管理テーブルの更新を要求し（図５のステップ９０２）、リプライを待つ。例えば、ファイルサーバ２４０からの要求を受信したノード１００のファイルサーバ１４０は、受信した論理ディスク管理テーブル２７０の更新情報に基づいて、ノード１００の論理ディスク管理テーブル１７０のＩＯパス“（２００，２９０，１１）”に対応する状態フラグ１７５を「使用不可」状態に、ＩＯパス“（３００，３９０，１２）”に対応する状態フラグ１７９を「使用中」状態に更新する（図５のステップ９０６）。この更新の後、ファイルサーバ１４０は、ファイルサーバ２４０に論理ディスク管理テーブル３７０の更新終了の通知を送る（図５のステップ９０７）。ファイルサーバ２４０が、全リモートノードのファイルサーバから論理ディスク管理テーブルの更新終了の通知を受信すれば（図５のステップ９０３）、システムを構成するすべてのノードの論理ディスク管理テーブルの更新が完了したことになる。

【００６９】リクエストの保留解除及び転送の処理（ステップ１００７）

この処理について、図１０を参照して説明する。ファイルサーバ２４０は、切り替え先のノードのファイルサーバ３４０にリクエストの保留を解除する要求を送る。この要求を受けたファイルサーバ３４０は、ステップ１００３で行ったＩＯアクセス要求の保留を解除し、保留していたＩＯアクセス要求の処理を行い、通常運用時の処理を開始する。また、ファイルサーバ２４０は、ステップ１００３で行ったＩＯアクセス要求の保留を解除し、保留していたＩＯアクセス要求のうち、障害発生ＩＯパスを指定したＩＯアクセス要求を、切り替え先のＩＯパスを指定したＩＯアクセス要求に変換した後、保留中のすべてのＩＯアクセス要求を、切り替え先のノードのファイルサーバ３４０に転送する。図１０に示す例では、ファイルサーバ２４０は、ＩＯパス“（２００，２９０，１１）”を指定したリード要求を、ＩＯパス“（３００，３９０，１２）”を指定したリード要求に変換し、前述の要求と論理ディスクＩＤ“１２３”を指定したライト要求とをノード３００のファイルサーバ３４０に転送している。転送されたＩＯアクセス要求は、ファイルサーバ３４０によって処理される。

【００７０】マウント構成ファイルの更新の処理（ステップ１００８）

最後に、ファイルサーバ２４０は、高信頼ディスク装置３０が接続されているノード１００のファイルサーバ１４０に障害発生ＩＯパス“（２００，２９０，１１）”が「使用不可」状態になったことをマウント構成ファイル２４に記載するように要求し、通常運用時の処理を開始する。この要求を受けたファイルサーバ１４０は、高信頼ディスク装置３０上のマウント構成ファイル２４を参照し、障害発生ＩＯパス“（２００，２９０，１１）”の使用可否情報５２を“unavailable”（使用不可）に書き換える。以上により、ＩＯパスの切り替え処理が終了する。

【００７１】論理ディスク管理テーブルの更新の処理（ステップ１０１１）

ステップ１０１０のチェックで、ＩＯパスの切り替え処理を行う必要がなかった場合、障害発生ノードのファイルサーバ２４０は、ステップ１００６の処理と同様の手順でシステム全体の論理ディスク管理テーブルを更新する。但し、障害発生ＩＯパス“（２００，２９０，１１）”の状態フラグを「待機中」から「使用不可」に書き換える処理だけを行う。システム全体の論理ディスク管理テーブルの更新が終了した後前述したステップ１００８の処理に進む。

【００７２】図１１はＩＯパスが障害から復旧したとき、ＩＯパスをシステムに復旧させる処理手順を説明するフローチャートであり、これについて説明する。ここでは、物理ディスク装置１０のディスクコントローラ１１の障害などの原因により「使用不可」状態になっていたＩＯパス“（２００，２９０，１１）”がディスクコントローラ１１の交換などによって再び使用可能になったとき、システムに上記のＩＯパスを復旧させる方法を例に説明する。また、ここでは、ＩＯパスの復旧処理中に使用中のＩＯパスに障害が発生することはないと仮定する。

【００７３】（１）障害が発生したディスクコントローラの交換等により、今まで使用不可能となっていたＩＯパス“（２００，２９０，１１）”が使用可能な状態になると、システム管理者は、管理用のプログラムを使って、このＩＯパスをシステムに復旧させる要求を、高信

頼ディスク装置が接続されているノード１００のファイルサーバ１４０に送信する。ファイルサーバ１４０は、この要求を受信する（ステップ６０１）。

【００７４】（２）復旧要求を自したファイルサーバ１４０は、論理ディスク管理テーブル１７０を参照して、前述のＩＯパス“（２００，２９０，１１）”の状態フラグ１７５を「使用不可」状態から「待機中」状態に更新する。また、ファイルサーバ１４０は、論理ディスク管理テーブル１７０の更新が終了したら、全ての稼働中のノードのファイルサーバと通信を行い、全ノードの論理ディスク管理テーブルを論理ディスク管理テーブル１７０と同じ内容にする。この処理は、図７によるＩＯパスの切り替えのフローにより説明したステップ１００６での処理と同様な処理により行われる（ステップ６０２）。

【００７５】（３）そして、ファイルサーバ１４０は、高信頼ディスク装置３０上のマウント構成ファイル２４を参照し、前述のＩＯパス“（２００，２９０，１１）”の使用可否情報５２を“unavailable”（使用不可）から“available”（使用可）に変更する。前述の処理により、ＩＯパス“（２００，２９０，１１）”を「待機中」状態としてシステムに復旧させることができる（ステップ６０３）。

【００７６】前述した本発明の実施形態は、ファイル管理テーブル２６０及びバッファキャッシュ２７０をノード２００からノード３００に転送するとして説明した（図７のステップ１００５）が、これは次のような理由による。すなわち、物理ディスク装置へのアクセスは、ローカルノードからのアクセスでもリモートノードからのアクセスでも、最終的に、その物理ディスク装置が接続されたノードのファイル管理テーブル及びバッファキャッシュを経由して行われる。従って、物理ディスク装置が接続されたノードは、そのノード（ローカルノード）で動作するプログラムに関係するファイル管理テーブル及びバッファキャッシュの他に、リモートノードで動作するプログラムに関係するファイル管理テーブル及びバッファキャッシュを持つ。前述した本発明の実施形態に示したようなＩＯパス切り替え処理は、物理ディスク装置が接続されているノードがノード２００からノード３００に切り替わるので、ノード３００がノード２００に代わって、ノード２００が保持していたファイル管理テーブル２６０及びバッファキャッシュ２８０を持つ必要がある。そこで、ＩＯパス切り替え処理時にファイル管理テーブルやバッファキャッシュをノード３００に転送するようにしている。このとき、ｄｉｒｔｙなデータのみを転送するようにして、データの転送量をなるべく少なく済むようにしている。

【００７７】また、前述した本発明の実施形態は、物理ディスク装置１０、２０を共にノード２００から使用しているときに、ＩＯインターフェイス２９０に障害が発生した場合、ＩＯパス（２００，２９０，１１）及び（２００，２９０，２１）の両方が使用できなくなるが、この場合、ディスクドライバ２５０が、各々のＩＯパスに対して障害検出を行い、各々のＩＯパスに対して前述の各ステップで示されるＩＯパスの切り替え処理を行うようにすればよい。また、ディスクドライバ２５０がＩＯインターフェイス２９０で障害が起こったことを検出する機能を持つ場合、ステップ１００１で、ディスクドライバ２５０がファイルサーバ２４０にＩＯインターフェイス２９０の障害を通知し、ステップ１００２で、ファイルサーバ２４０が論理ディスク管理テーブル２７０を検索し、障害発生ＩＯインターフェイス番号“２９０”から、障害発生ＩＯパス（２００，２９０，１１）、（２００，２９０，２１）と対応する切り替え先のＩＯパスと論理ディスクＩＤを探し出し、これら２組のＩＯパスについて、前述の各ステップで示される切り替え処理を同時に行うようにしてもよい。

【００７８】前述した本発明の実施形態において、ノード２００が２つのＩＯインターフェイスを有し、物理ディスク装置１０がこれら２つのＩＯインターフェイスによってノード２００と接続されており、物理ディスク装置１０とノード２００との間のＩＯパスが２つ存在し、通常運用時これらのＩＯパスのうち１つを利用しているような場合、ディスクコントローラやＩＯインターフェイスの障害発生により、今まで使用していたＩＯパスが使用できなくなったとき、物理ディスク装置１０にアクセスするためのＩＯパスをもう片方のＩＯパスに前述したの方法で切り替えることができる。この場合、ステップ１００３でノード３００のファイルサーバ３４０がＩＯアクセス要求を保留する処理と、ステップ１００５でノード２００が持つバッファキャッシュ２８０及びファイル管理テーブル２６０をＩＯパス切り替え先のノード３００に転送する処理が不要となる。

【００７９】また、本発明は、物理ディスク装置にアクセスするためのＩＯパスが３つ以上存在する場合にも適用することができる。この場合、論理ディスク管理テーブル及びマウント構成ファイル２４の各エントリに３つ以上のＩＯパスの組を登録できるようにし、システムの起動時にファイルサーバ１４０がマウント構成ファイル２４に記載されたＩＯパスの組に対して、１つの論理ディスクＩＤを設定し、ＩＯパスと論理ディスクＩＤとの対応関係を論理ディスク管理テーブルに登録するようにすればよい。そして、この場合、通常運用時、複数のＩＯパスが「待機中」状態としてスタンバイするため、障害発生時のＩＯパスの切り替え処理を行う際に、複数の「待機中」状態のＩＯパスの中から切り替え先のＩＯパスを選択する必要がある。この切り替え先のＩＯパスの決定は、前述した実施形態におけるステップ１００２で障害を検出したノードのファイルサーバがそのノードの論理ディスク管理テーブルを検索し、障害発生ＩＯパス

名を含むエントリを見つけたときに、そのエントリのなるべく最初の方のフィールドに登録されている「待機中」状態のＩＯパスを切り替え先のＩＯパスとして選び出すことによって行うようにすればよい。また、論理ディスク管理テーブルに登録されている各ＩＯパス毎に使用時間（状態フラグが「使用中」状態となっていた時間）を上記論理ディスク管理テーブルに登録できるようにし、ＩＯパスの切り替え処理時、使用時間の短いＩＯパスに切り替えるようにしてもよい。これによって、複数のＩＯパスをまんべんなく使用することができる。

【００８０】さらに、本発明は、ＬＡＮ等のネットワークにより接続された疎結合計算機システムによるファイルシステムに対しても適用することができる。この場合、前述のノード番号の代わりにネットワークアドレスを使用すればよい。

【００８１】また、前述した本発明の実施形態において、ディスクキャッシュ１３をディスクコントローラ１２から制御し、ディスク装置１０に書き戻す機能を物理ディスク装置１０が持たない場合、ノード２００のディスクドライバ２５０が、ディスクキャッシュ１３に保存されたｄｉｒｔｙなキャッシュを少なくとも含むデータを予め保持しておいて、障害発生時、前述のステップ１００４でディスクドライバ２５０がディスクドライバ３５０と通信を行い、ｄｉｒｔｙなディスクキャッシュを少なくとも含むようなデータをノード２００からノード３００に転送し、ディスクコントローラ１２を通してディスク装置１０に書き戻すようにしてもよい。

【００８２】前述した本発明の実施形態は、ＩＯパス切り替え処理中、障害発生ノード及び切り替え先のノードに送信されてきたＩＯアクセス要求は、保留するようにしていたが、ＩＯアクセス要求を保留しないようにすることもできる。以下、この場合のファイルサーバの動作について図面により説明する。

【００８３】図１２はＩＯパス切り替え時の障害発生ノードの処理動作の他の例について説明するフローチャート、図１３は障害発生ノード以外のノードの処理動作の他の例を説明するフローチャートである。以下、障害発生ノードがノード２００、切り替え先のノードがノード３００の場合を例として、図１２、図１３に示すフローを参照して、ＩＯパス切り替え処理中に各ノードに送信されてきたＩＯアクセス要求の処理の方法を説明する。まず、障害発生ノードのファイルサーバの動作を図１２のフローにより説明する。

【００８４】（１）障害発生ノードのファイルサーバ２４０は、ＩＯパス切り替え処理中に、ＩＯアクセス要求を受信すると、その要求が他のノードであるリモートノードからの要求が否かを判定する（ステップ７０１、７０２）。

【００８５】（２）ステップ７０２の判定で、受信したＩＯアクセス要求がローカルノード（自ノード）のユーザアプリケーション２３０からのものであると判定すると、ファイルサーバ２４０は、前述した実施形態で説明したと同様に、ＩＯパス切り替え処理の間、その要求を保留する。この要求は、ＩＯパスの切り替え処理終了時に、切り替え先のノードに送信される（ステップ７０３）。

【００８６】（３）ステップ７０２の判定で、受信したＩＯアクセス要求がリモートノードからのものであると判定すると、ファイルサーバ２４０は、その要求に対してリプライを返さずに無視する（ステップ７０４）。

【００８７】次に、障害発生ノード以外のノードのファイルサーバの動作を図１３に示すフローを参照して説明する。障害発生ノード以外のファイルサーバは、基本的に図４により説明した通常運用時と同様の動作をするので、ここでは図４の処理と重なる部分については説明を省略する。

【００８８】（１）障害発生ノード以外のファイルサーバがＩＯパス切り替え中に障害発生ノード（ノード２００）に送信したＩＯアクセス要求はタイムアウトとなる（ステップ８０８）。

【００８９】（２）ＩＯアクセス要求がタイムアウトになったら、ＩＯアクセス要求を送信したファイルサーバは、一定時間（例えば１秒）待った後、論理ディスク管理テーブルを参照して、論理ディスクＩＤからＩＯパス名を求める処理から処理をやり直す。このとき、ＩＯパスの切り替え処理が終了していれば、全ノードの論理ディスク管理テーブルが更新されているので、ステップ８０４の処理によって切り替え先のＩＯパスが求まる（ステップ８０４）。

【００９０】（３）ＩＯアクセス要求を送信しようとしているファイルサーバは、求められたＩＯパス名が含むノードがローカルノードであるか否かを判定し、切り替え先のＩＯパス名が含むノードがローカルノードでなかった場合、ＩＯアクセス要求を切り替え先のノード（ノード３００）に送信する（ステップ８０５、８０６）。

【００９１】（４）ステップ８０５の判定で、切り替え先のＩＯパスがローカルノードであれば、ＩＯアクセス要求を送信しようとしているファイルサーバは、ＩＯアクセス要求をローカルノードのディスクドライバに送信する（ステップ８０７）。

【００９２】前述したステップ８０４の処理において、もし、論理ディスクＩＤからＩＯパス名を求めなおしたときに、ＩＯパスの切り替え処理が終了していない場合、ＩＯアクセス要求は、障害発生ノード（ノード２００）に送信され、上記ＩＯアクセス要求は再びタイムアウトとなり、ＩＯアクセス要求が成功するまで前述した処理が繰り返される。

【００９３】この方法を使用することにより、図７により説明したステップ１００３のリクエストの保留処理でリモートノードからのアクセス要求を保留する必要がな

くなるので、ＩＯアクセス要求を保留するためのメモリを節約することができる。また、ＩＯアクセス要求の再送回数に制限（例えば５回）を設け、もし制限回数だけ再送を行ってもタイムアウトになり続ければ、そのＩＯアクセス要求をエラーとしてもよい。また、ＩＯパス切り替え処理中、障害発生ノードのファイルサーバ２４０は、リモートノードからのＩＯアクセス要求を無視するかわりに、「ＩＯパス切り替え処理中なので、ＩＯアクセス要求を処理できない」という意味の通知をアクセス要求を送信したリモートノードのファイルサーバに送信するようにしてもよい。これにより、リモートノードのファイルサーバは、ＩＯパスで障害が発生した場合とノード２００で障害が発生した場合とを区別することができるようになる。

【００９４】前述までに説明した本発明の第１の実施形態によるＩＯパス切り替え方法は、ノード２００でＯＳの障害が発生したとき、ネットワーク１を通じてバッファキャッシュ２８０やファイル管理テーブル２６０をノード３００に転送することができなくなるため、同じ方法でＩＯパスの切り替えを行うことは不可能である。

【００９５】これを解決するため、本発明は、バッファキャッシュ２８０やファイル管理テーブル２６０をノード３００に転送するための専用のハードウェアを使う方法を取ることができる。以下、これを第２の実施形態として説明する。

【００９６】図１４は本発明の第２の実施形態によるディスクキャッシュの書き戻しの処理とバッファキャッシュの転送の処理とを説明する図である。

【００９７】本発明の実施形態におけるＩＯパス切り替え処理の手順は、前述までに説明した第１の実施形態の場合の図７に示すフローと同様に行われる。但し、第２の実施形態では、ステップ１００３及びステップ１００７の処理は行わない。そして、図１４には、ステップ１００４でのディスクキャッシュの書き戻しの処理とステップ１００５でのバッファキャッシュの転送の処理についてしめしている。

【００９８】図１４において、メモリアクセス手段２９９（３９９）は、ノード２００（３００）に付属しており、メモリアクセス手段２９９とメモリアクセス手段３９９とは専用通信線２によって互いに接続されている。メモリアクセス手段２９９は、ノード２００でＯＳの障害が発生しノード２００上で動作するプログラムの全てが停止した場合にも、メモリ２２０にアクセスし、その内容を専用通信線２を使用してメモリアクセス手段３９９との通信によりノード３００に送信することが可能なハードウェアである。

【００９９】通常運用時、図１４に示す各ノードのファイルサーバは、図１３により説明した動作を行う。ここで例えば、ノード２００でＯＳの障害が発生したとすると、あるファイルサーバがノード２００に送信したＩＯアクセス要求のリプライが戻ってこないので、ＩＯアクセスを送信したファイルサーバは、上記ＩＯアクセス要求をタイムアウトにする（ステップ８０８）。ファイルサーバは、一定時間待った後、ローカルノードの論理ディスク管理テーブルを参照し、論理ディスクＩＤからＩＯパスを求める処理から処理の再実行を行うことになる（ステップ８０４）。ＩＯパス切り替え処理中、前述の要求は、障害発生ノード（ノード２００）に送信されタイムアウトとなるが、ＩＯパス切り替え終了後、要求は切り替え先のノードに送信される。

【０１００】以下、ノード２００で障害が発生しノード２００で動作する全てのプログラムが停止した場合に、物理ディスク装置１０にアクセスするためのＩＯパスを（２００，２９０，１１）から（３００，３９０，１２）に切り替えるものとして、その処理を図１、図２、図１４を併用しながら図７に示すフローを参照して説明する。

【０１０１】障害検出の処理（ステップ１００１）

ノード２００で障害が発生すると、ノード２００は、リクエストを一切受け付けなくなる。従って、ノード２００にＩＯアクセス要求を送信したリモートノードのファイルサーバは、ＩＯアクセス要求をタイムアウトとする。ＩＯアクセス要求を送信したファイルサーバは、このタイムアウトによってノード２００で障害が発生したことを検出する。前述したように、ＩＯアクセス要求を送信したファイルサーバは、ＩＯ処理要求がタイムアウトになったらその要求を再送するので、何度も障害発生ノード（ノード２００）に上記要求を再送し、そのたびに要求をタイムアウトにする可能性がある。上記ファイルサーバは、あるノードへの要求が最初にタイムアウトになったとき、次のステップ１００２の処理に進み、２回目以降、ステップ１００２以降の処理は行わない。

【０１０２】切り替え対象ＩＯパスの検索の処理（ステップ１００２）

ＩＯアクセス要求を送信したファイルサーバは、ローカルノードの論理ディスク管理テーブルを参照し、障害が発生したノードのノード番号“２００”から障害発生ＩＯパス名と切り替え先のＩＯパス名とを探し出し、切り替え先のＩＯパスが含むノード番号に対応するノード（切り替え先のノード）のファイルサーバに、障害発生ＩＯパスから切り替え先のＩＯパスにＩＯパスを切り替えるように要求する。切り替え先のノードがローカルノード（自ノード）であれば、ＩＯアクセスを送信したファイルサーバは、直ちにＩＯパスの切り替えの処理を開始する。但し、障害発生ＩＯパスの状態フラグが「待機中」状態の場合（ステップ１０１０）、ＩＯパスの切り替え処理は必要なくステップ１０１１の処理に進む。例えば、ノード１００のファイルサーバ１４０がノード２００のファイルサーバ２４０に送信したＩＯ処理要求がタイムアウトとなった場合、ファイルサーバ１４０は、

図２に示した論理ディスク管理テーブル１７０を検索し、ノード番号“２００”を含むエントリを探す。見つかったエントリには複数のＩＯパスが記載されているが、ノード番号“２００”を含むＩＯパス“（２００，２９０，１１）”が障害発生ＩＯパスであり、状態フラグが「待機中」状態でノード番号“２００”を含まないＩＯパス“（３００，３９０，１２）”が切り替え先のＩＯパスである。障害発生ＩＯパスの状態フラグ２７５が「使用中」状態であるので、ファイルサーバ１４０は、切り替え先のノード３００のファイルサーバ３４０に“（２００，２９０，１１）”から“（３００，３９０，１２）”にＩＯパスを切り替えるように要求する。もし、上記障害発生ＩＯパスの状態フラグが「待機中」状態であれば、ＩＯパスの切り替え処理は必要なく、ステップ１０１１の処理に進む。

【０１０３】前述した検索処理で、切り替え処理を行うＩＯパスの組が複数個見つかった場合、障害を検出したファイルサーバは、ＩＯパス毎に対応する切り替え先のノードのファイルサーバにＩＯパスの切り替え要求を送信する。但し、複数のＩＯパスの切り替え要求を１つのノードに送る必要がある場合、それらのＩＯパスの切り替え要求を一括して送り、切り替え先のノードのファイルサーバが、それらのＩＯパスの切り替え処理を同時に行う。例えば、物理ディスク装置１０と物理ディスク装置２０とをノード２００から使用していた場合、ノード２００の障害を検出したファイルサーバは、ノード３００のファイルサーバ３４０に上記２つの物理ディスク装置にアクセスするための２組のＩＯパスを切り替える要求を発行し、ファイルサーバ３４０は、前述した２組のＩＯパスの切り替え処理を同時に行う（ステップ１００４～１００８）。

【０１０４】ディスクキャッシュの書き戻しの処理（ステップ１００４）

障害発生ＩＯパス“（２００，２９０，１１）”から切り替え先のＩＯパス“（３００，３９０，１２）”にＩＯパスを切り替えるように要求されたファイルサーバ３４０は、ＩＯパスの切り替えモードに入り、その後再び同じＩＯパス切り替え要求が送られてきても受理しない。これによって、ＩＯパスの切り替え処理が二重に行われることを防止する。このステップの処理の後の処理内容は、第１の実施形態の場合と同様に行われる。ファイルサーバ３４０は、図１４に示すように、ディスクドライバ３５０にディスクキャッシュの書き戻し要求を送信することにより、ディスクキャッシュ１３の内容をディスク領域に書き戻して、ディスクコントローラ１１を物理ディスク装置から切り放す。

【０１０５】バッファキャッシュの移動の処理（ステップ１００５）

ファイルサーバ３４０は、次に、図１４に示すように、メモリアクセス手段３９９に、障害が発生したノード２００のファイル管理テーブル２６０とバッファキャッシュ２８０との内容をローカルノード（ノード３００）に転送するように要求する。メモリアクセス手段３９９は、メモリアクセス手段２９９と通信を行い、専用通信線２を介して、ｄｉｒｔｙなバッファキャッシュ２８０及びｄｉｒｔｙなファイル管理テーブル２６０の内容をノード３００のファイルサーバ３４０に転送する。ファイルサーバ３４０は、ノード３００上のファイル管理テーブル３６０及びバッファキャッシュ３８０にメモリアクセス手段３９９から送られてきたデータをマージする。マージされたデータは、ＩＯパスの切り替え終了後、ファイルサーバ３４０によって切り替え先のＩＯパスから物理ディスク装置１０に書き込まれる。また、これらデータは、読み出し用のキャッシュとしても使われる可能性もある。

【０１０６】論理ディスク管理テーブルの更新の処理（ステップ１００６）

データの転送処理が終了した後、ファイルサーバ３４０は、論理ディスク管理テーブル３７０に登録されているＩＯパスの状態フラグを、障害発生ＩＯパス“（２００，２９０，１１）”について、「使用不可」状態に、切り替え先のＩＯパス“（３００，３９０，１２）”について、「使用中」状態に登録し直す。ファイルサーバ３４０は、論理ディスク管理テーブル３７０の更新の終了後、第１の実施形態の場合と同様な方法により、全ての稼働中のノードのファイルサーバと通信を行うことにより、全ての稼働中のノードの論理ディスク管理テーブルに登録されている、障害発生ＩＯパスの状態フラグを「使用不可」状態に、切り替え先のＩＯパスの状態フラグを「使用中」状態に更新する。

【０１０７】マウント構成ファイルの更新の処理（ステップ１００８）

ファイルサーバ３４０は、全ての稼働中のノードの論理ディスク管理テーブルの更新が終了した後、高信頼ディスク装置３０が接続されているノード１００のファイルサーバ１４０に、ＩＯパス“（２００，２９０，１１）”が「使用不可」状態になったことをマウント構成ファイル２４に記載するように要求し、ＩＯパスの切り替えモードから抜け、通常運用時の処理を開始する。前述の要求を受けたファイルサーバ１４０は、「使用不可」状態となったＩＯパス“（２００，２９０，１１）”の使用可否情報５２を“available”（使用可）から“unavailable”（使用不可）に更新する。以上によりＩＯパスの切り替え処理が終了する。

【０１０８】論理ディスク管理テーブルの更新の処理（ステップ１０１１）

ステップ１０１０で、障害発生パスが「待機中」状態にあると判定され、ＩＯパスの切り替え処理を行う必要がない場合、ステップ１００１の処理で障害を検出したファイルサーバは、ステップ１００６の処理と同様の手順

でシステム全体の論理ディスク管理テーブルを更新する。但し、障害発生ＩＯパスの状態フラグを「使用不可」に書き換える処理だけを行う。システム全体の論理ディスク管理テーブルの更新が終了した後、前述のファイルサーバがファイルサーバ１４０に対してマウント構成ファイルの更新を要求し、この要求を受けたファイルサーバ１４０は、ステップ１００８の処理を行う。

【０１０９】図１５は本発明の第３の実施形態によるファイルシステムの構成を示すブロック図、図１６は本発明の第３の実施形態におけるマウント構成ファイルの具体的な構成例を説明する図であり、図１５における符号は図１の場合と同一である。図１５に示す本発明の第３の実施形態は、同一のファイルを物理ディスク装置１０と物理ディスク装置２０とに二重化（ミラーリング）して記録する例である。

【０１１０】図示本発明第３の実施形態において、マウント構成ファイルの１つのエントリには、図１６に示すように、物理ディスクにアクセスするためのＩＯパス名５１、５３、各ＩＯパスの使用可否情報５２、５４、マウントポイント５５が記載されている。この第３の実施形態は、マウントポイントの１つのエントリに記載されたＩＯパスからアクセスされる物理ディスク装置にファイルが多重化して記録される。従って、前述のＩＯパスからアクセスされる物理ディスク装置は異なるものである必要がある。図１６に示す例では、／mnt ディレクトリ以下のディレクトリに格納されたファイルは、ＩＯパス“（２００，２９０，１１）”、“（３００，３９０，２２）”からアクセスされる物理ディスク装置（物理ディスク装置１０、２０）にミラーリングされる。このような指定方法を採用することにより、システム管理者が論理ボリュームの設定を行う必要がなくなる。

【０１１１】システム立ち上げ時、ファイルサーバ１４０は、マウント構成ファイル２４を読み込んで、第１の実施形態の場合と同様の手順で、全てのノードの論理ディスク管理テーブルを設定する。但し、第３の実施形態では、ファイルサーバ１４０は、マウント構成ファイル２４の使用可否情報に“available”（使用可）と記載されているすべてのＩＯパスについて、論理ディスク管理テーブルの対応する状態フラグに「使用中」と登録する。

【０１１２】次に、通常運用時のファイルサーバの動作を、ノード１００のユーザアプリケーション１３０がファイルＩＤ“１００”を指定したファイルアクセス要求をファイルサーバ１４０に発行した場合を例に、図１５、図１６を参照し、図６に示すフローに基づいて説明する。なお、ファイル管理テーブルの設定は図２、論理ディスク管理テーブルの設定は図１６に示すようになっているものとする。

【０１１３】（１）ファイルサーバ１４０は、ユーザアプリケーション１３０からファイルＩＤを指定したアクセス要求を受けると、その要求がリモートノードからの要求であるか否かを判定し、自ノードからの要求である場合、ファイル管理テーブル１６０を検索し、ファイルＩＤ“１００”から論理ディスクＩＤ“１２３”を求める（ステップ５０１～５０３）。

【０１１４】（２）そして、ファイルサーバ１４０は、論理ディスク管理テーブル１７０を検索し、論理ディスクＩＤ“１２３”から状態フラグが「使用中」状態のＩＯパス名“（２００，２９０，１１）”、“（３００，３９０，２２）”を求める（ステップ５０４）。

【０１１５】（３）アクセス要求がライト要求の場合は、前述の両方のＩＯパスに対して同一内容の書き込みを行う。このため、ファイルサーバ１４０は、前記２つのＩＯパス名が含むノードがローカルノードか否かを判定し、ローカルノードでない場合、すなわちリモートノードである場合、２つのＩＯパスが含むノード番号に対応するノード（ノード２００、ノード３００）のファイルサーバ２４０、３４０にＩＯパス名を指定したライト要求を送信する（ステップ５０５、５０６）。

【０１１６】（４）ステップ５０５での判定が、ノードがローカルノードであった場合、ローカルノードのディスクドライバにＩＯパスを指定したライト要求を送信する（ステップ５０７）。

【０１１７】図１５に示す例の場合、前述の処理で、ファイルサーバ１４０は、ファイルサーバ２４０にＩＯパス“（２００，２９０，１１）”を指定したライト要求を送信し、ファイルサーバ３４０にＩＯパス“（３００，３９０，２２）”を指定したライト要求を送信する。これらのライト要求を受信したファイルサーバ２４０、３４０は、それぞれのノードのディスクドライバにＩＯパスを指定したライト要求を送信する。

【０１１８】受信したアクセス要求がリード要求の場合、ファイルサーバ１４０は、前述したＩＯパスのうちで、論理ディスク管理テーブルの最も最初のフィールドに登録されていたＩＯパス“（２００，２９０，１１）”を使用してアクセスを行う。もし、ＩＯパスの障害などの理由により、このＩＯパスを使用してアクセスすることができない場合、順に次のフィールドに登録されているＩＯパスを使用してアクセスを試みる。また、前述のＩＯパスの中で、ローカルノードのノード番号を含むものがあれば、そのＩＯパスを最初に使うようにしてもよい。このように、なるべくリモートアクセスを減らすことによって、ネットワークの負荷を減らすことができる。リード処理に使用するＩＯパスが決定した後の処理は、ライト要求の場合と同様である。

【０１１９】次に、障害発生時、障害が発生したＩＯパスを切り放す処理を説明する。ここでは、ディスクコントローラやＩＯインターフェイスの障害により、ノード２００に接続されていた物理ディスク装置２０にアクセスするためのＩＯパス“（２００，２９０，１１）”が

使用不可能になったものとして説明する。

【０１２０】障害の発生により、ＩＯパス“（２００，２９０，１１）”が使用できなくなった場合、ノード２００のデバイスドライバ２５０は、このＩＯパスの障害を検出し、障害発生をファイルサーバ２４０に通知する。

【０１２１】この通知を受けたファイルサーバ２４０は、論理ディスク管理テーブル２７０を更新し、障害発生ＩＯパスの状態フラグを「使用不可」状態にする。ファイルサーバ２４０は、図５に示したフローによる方法により、全てのリモートノードのファイルサーバと通信を行い、全てのノードの論理ディスク管理テーブルを論理ディスク管理テーブル２７０と同一の内容に更新する。

【０１２２】最後に、ファイルサーバ２４０は、高信頼ディスク装置３０が接続されたノード１００のファイルサーバ１４０に、障害発生ＩＯパス“（２００，２９０，１１）”が「使用不可」状態になったことを、マウント構成ファイル２４に記載するように要求する。この要求を受けたファイルサーバは、マウント構成ファイル２４を更新し、上記障害発生ＩＯパスの使用可否情報を“unavailable”（使用不可）に書き換える。以上によりＩＯパスの切り離しが終了する。

【０１２３】ＩＯパスの切り離し処理中に、あるノードのファイルサーバ（例えば、ファイルサーバ１４０）が、ファイルサーバ２４０に前述の障害発生ＩＯパスを指定したアクセス要求を送るとその要求は失敗する。しかし、ライト処理の場合、データは、同時に複数の物理ディスク装置に書き込まれるので、アクセス可能な物理ディスク装置（物理ディスク装置２０）の方に無事に記録されている。また、リード処理の場合、アクセス要求を行ったファイルサーバは、アクセスに失敗したら別のＩＯパス“（３００，３９０，２２）”を指定したＩＯアクセス要求をファイルサーバ３４０に送信する。このため、データは、アクセス可能な物理ディスク装置から無事に読み込まれる。従って、ＩＯパス切り替え中もユーザは、それを意識することなくファイルにアクセスすることができる。

【０１２４】前述した本発明の実施形態において、ノード２００で障害が発生したことにより、ＩＯパス“（２００，２９０，１１）”が使用できなくなった場合、ノード２００にＩＯアクセス要求を送信したリモートノードのファイルサーバが、送信したアクセス要求のタイムアウトによってノード２００の障害を検出し、障害を検出したこのファイルサーバが上記のＩＯパスの切り離し処理を行うようにすればよい。

【０１２５】また、前述した本発明の実施形態において、論理ディスク管理テーブルに、論理ディスクの使用方法（切り替え、ミラーリングなど）を指定するためのディスクタイプ情報を論理ディスクＩＤ毎に登録できるようにし、マウント構成ファイル２４に上記ディスクタイプ情報を登録できるようにし、システム起動時にファイルサーバ１４０がマウント構成情報２４に記載されたディスクタイプ情報を、論理ディスク管理テーブルのディスクタイプ情報に登録し、通常運用時及び障害発生時、ファイルサーバが論理ディスク管理テーブルのディスクタイプ情報によって、ディスクタイプを判別し各ディスクタイプ毎の処理を行うようにすることもできる。例えば、図１５に示す例の場合、マウント構成ファイル２４には“（（２００，２９０，１１） available）（（３００，３９０，２２） available） /mnt mirror”と記載する。“mirror”は、前述２つのＩＯパスからアクセスされる物理ディスク装置に対して、ミラーリングを行うことを示す。ファイルサーバ１４０は、起動時に前述のエントリを読み込んで、ディスクタイプが「ミラーリング」であることを判別し、論理ディスク管理テーブルの対応するディスクタイプ情報に、「ミラーリング」であることを登録する。通常運用時、ファイルサーバは、論理ディスク管理テーブルのディスクタイプ情報を参照して、前述のＩＯパスの組が「ミラーリング」を行うものであることを判別すると、前述した実施形態により説明した「ミラーリング」の処理を行う。ディスクタイプが「切り替え」の場合も同様である。これにより、ＩＯパスの切り替えとミラーリングをシステムで共存させることができる。

【０１２６】前述した本発明の第３の実施形態は、ファイルのミラーリングを行うものとして説明したが、論理ディスク管理テーブルの１つのエントリに登録されたＩＯパスからアクセスされる物理ディスク装置に、ファイルを分散して記録するようにすれば、ファイルのストライピングを行うことができる。

【０１２７】

【発明の効果】以上説明したように本発明によれば、ＩＯパス切り替え・復旧処理のためにかかる時間を短縮することができ、また、ＩＯパス切り替え時にファイルの整合性のチェックを不要にすることができる。また、本発明によれば、ＩＯパスの切り替え・切り離し処理が発生しても、一般ユーザはそれを意識することなく作業を続けることができる。さらに、本発明によれば、ＩＯパス切り替え・切り離し処理後あるいは障害発生ＩＯパス復旧後、システムを再起動する際にシステム管理者がマウント構成ファイルを設定しなおす必要をなくすことができ、システム管理者の負担を軽減することができる。

【図面の簡単な説明】

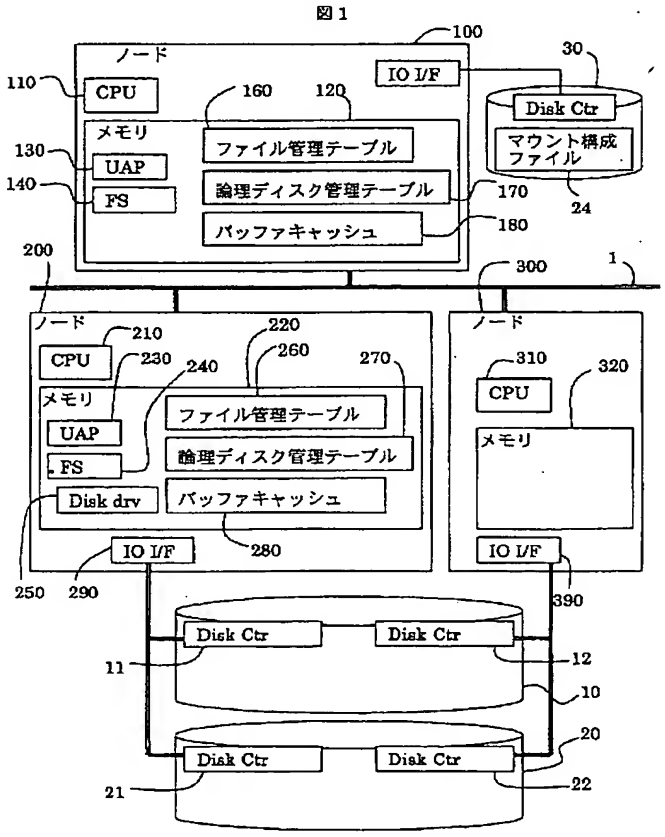
【図１】本発明の第１の実施形態によるファイルシステムの構成を示すブロック図である。

【図２】システム内に設けられる各種のテーブルの具対的な構成例を説明する図である。

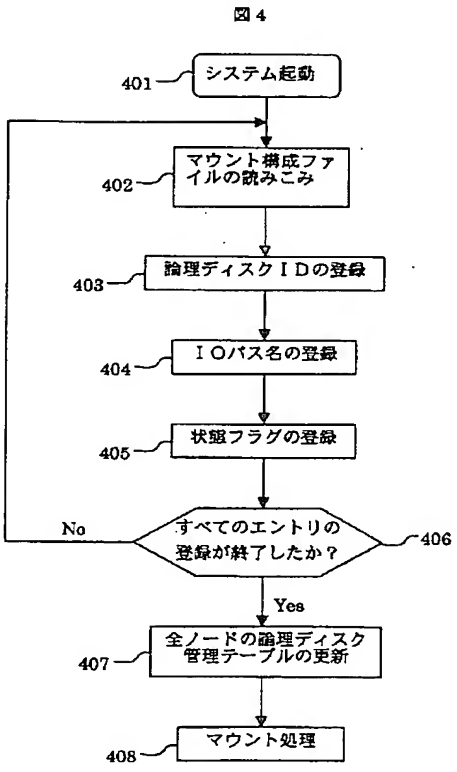
【図３】マウント構成ファイルの具体的な構成例を説明する図である。

【図４】システムの起動時のファイルサーバの処理動作を説明するフローチャートである。
【図５】システム全体のノードの論理ディスク管理テーブルを更新する処理動作を説明するフローチャートである。
【図６】通常運用時のファイルサーバの処理動作を説明するフローチャートである。
【図７】ＩＯパスの切り替えの処理動作を説明するフローチャートである。
【図８】ＩＯパスに障害が発生しＩＯパスの切り替えを行う処理について説明する図（その１）である。
【図９】ＩＯパスに障害が発生しＩＯパスの切り替えを行う処理について説明する図（その２）である。
【図１０】ＩＯパスに障害が発生しＩＯパスの切り替えを行う処理について説明する図（その３）である。
【図１１】ＩＯパスが障害から復旧したとき、ＩＯパスをシステムに復旧させる処理手順を説明するフローチャートである。
【図１２】ＩＯパス切り替え時の障害発生ノードの処理動作の他の例について説明するフローチャートである。
【図１３】障害発生ノード以外のノードの処理動作の他の例を説明するフローチャートである。
【図１４】本発明の第２の実施形態によるディスクキャッシュの書き戻しの処理とバッファキャッシュの転送の処理とを説明する図である。
【図１５】本発明の第３の実施形態によるファイルシステムの構成を示すブロック図である。
【図１６】本発明の第３の実施形態におけるマウント構成ファイルの具体的な構成例を説明する図である。
【符号の説明】
１　ネットワーク
１０、２０、３０　物理ディスク装置
１１、１２、２１、２２　ディスクコントローラ
１３　ディスクキャッシュ
２４　マウント構成ファイル
１００、２００、３００　ノード
１１０、２１０、３１０　ＣＰＵ
１２０、２２０、３２０　メモリ
１３０、２３０　ユーザアプリケーション（ＵＡＰ）
１４０、２４０、３４０　ファイルサーバ（ＦＳ）
１６０、２６０　ファイル管理テーブ
１７０、２７０　論理ディスク管理テーブル
１８０、２８０、３６０　バッファキャッシュ
２５０、３５０　ディスクドライバ
２９０、３９０　ＩＯインタフェース

【図１】



【図４】



【図２】

図２

ファイル管理テーブル 260 (160)

261 (161) ファイルＩＤ	262 (162) 論理ディスク ID	263 (163) ファイル管理情報
100	123	1999/12/10　128byte ...
200	123	1999/12/12　500byte ...

論理ディスク管理テーブル 270 (170)

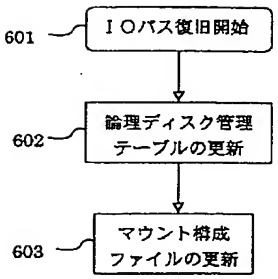
271 (171) 論理ディスク ID	272 (172) ノード番号	273 (173) IO I/F 番号	274 (174) Disk Ctr 番号	275 (175) 状態フラグ	276 (176) ノード番号	277 (177) IO I/F 番号	278 (178) Disk Ctr 番号	279 (179) 状態フラグ
123	200	290	11	使用可	300	390	12	待機中

バッファキャッシュ 280

281 論理ディスク ID	282 ブロック番号	283 キャッシュデータ
123	10	ＸＹＺ
123	11	ＡＢＣ

【図１１】

図 11



【図３】

図３

マウント構成ファイル２４

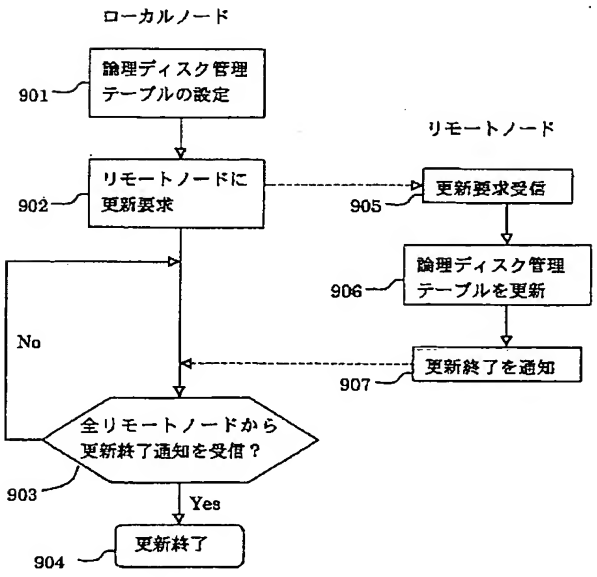
((ノード番号, IO I/F 番号, Disk Ctr 番号) 使用可否)　マウントポイント　<-- コメント行

((200, 290, 11) available) ((300, 390, 12) available) /mnt

51　52　53　54　55

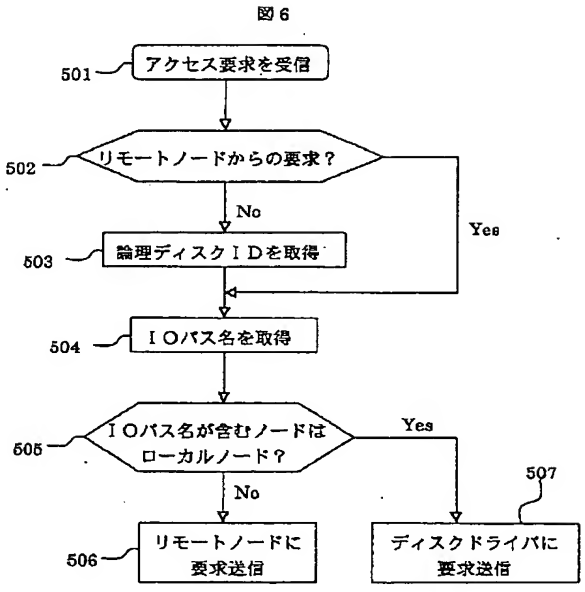
【図５】

図５



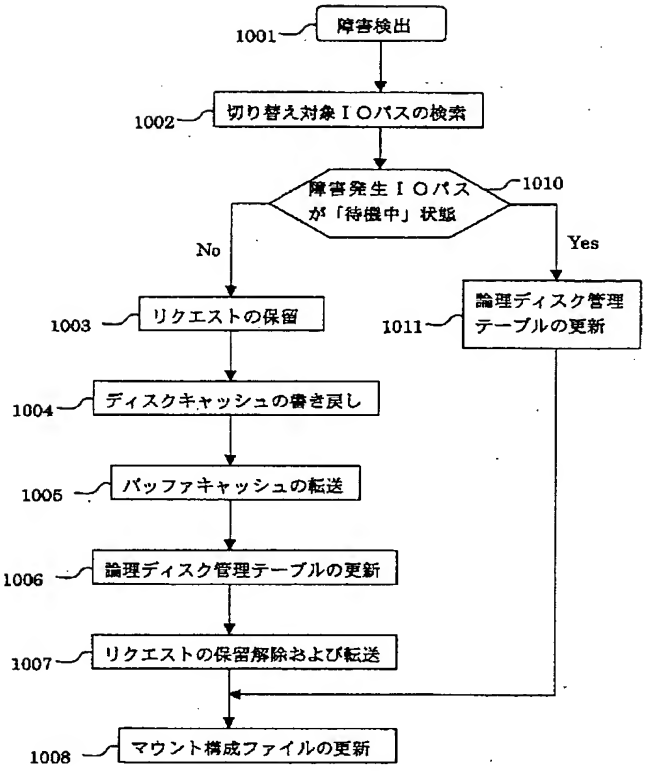
【図６】

図６



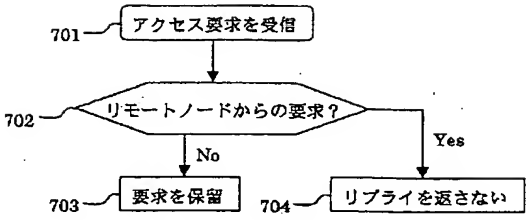
【図７】

図７



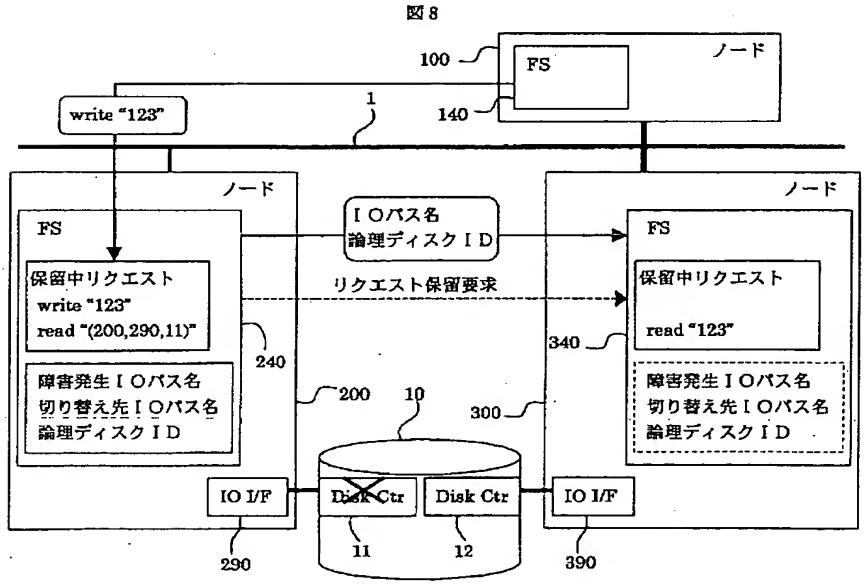
【図１２】

図１２



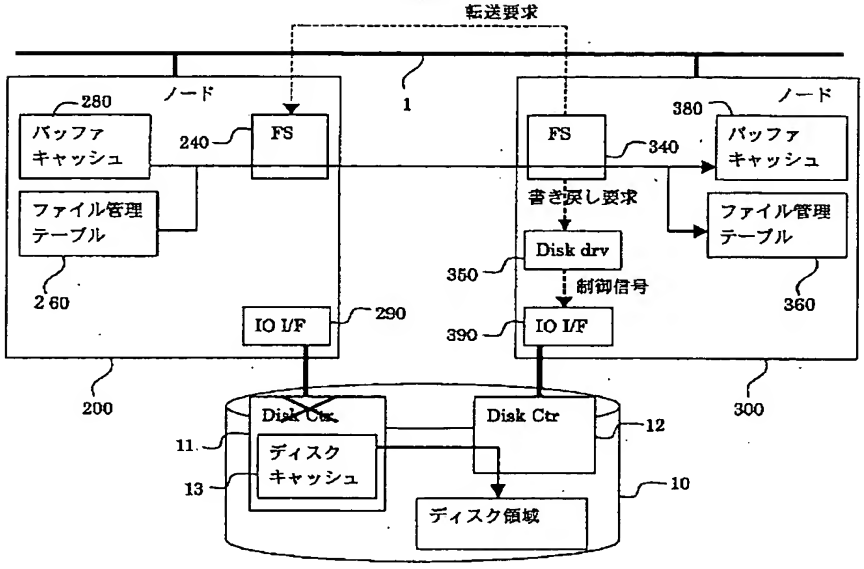
【図８】

図８



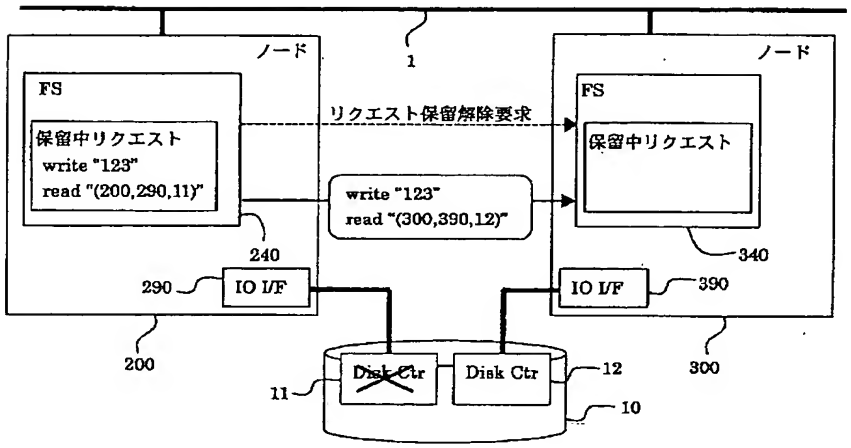
【図９】

図９


転送要求
1
ノード
280
バッファ
キャッシュ
240
FS
ファイル管理
テーブル
260
IO I/F
290
200
ノード
380
FS
340
バッファ
キャッシュ
書き戻し要求
ファイル管理
テーブル
Disk drv
350
制御信号
360
390
IO I/F
300
Disk Ctr
Disk Ctr
12
11
ディスク
キャッシュ
13
10
ディスク領域

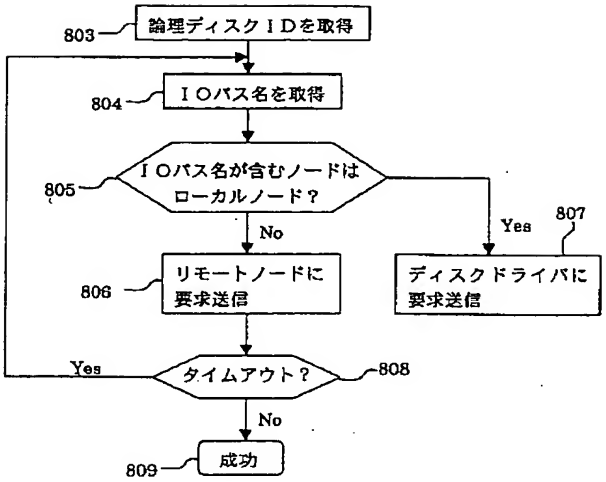
【図１０】

図１０


1
ノード
FS
保留中リクエスト
write "123"
read "(200,290,11)"
240
290
IO I/F
200
リクエスト保留解除要求
write "123"
read "(300,390,12)"
ノード
FS
保留中リクエスト
340
IO I/F
390
300
Disk Ctr
Disk Ctr
12
11
10

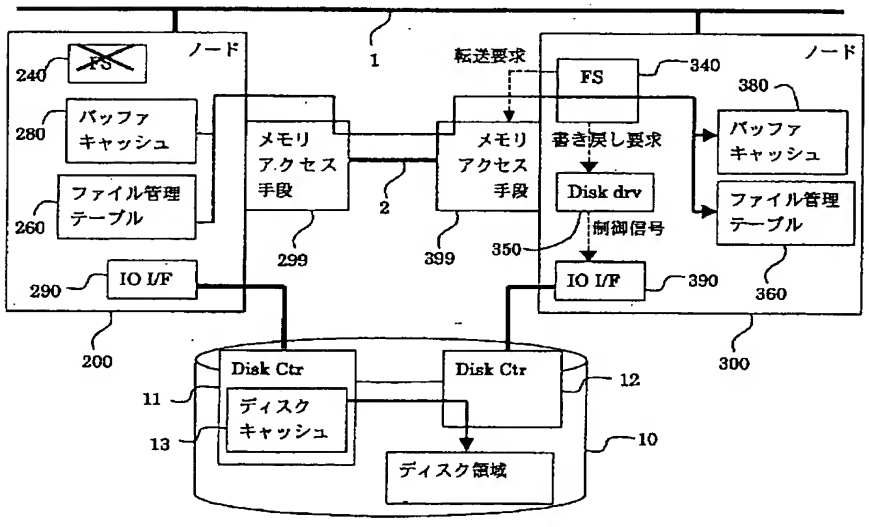
【図１３】

図 13



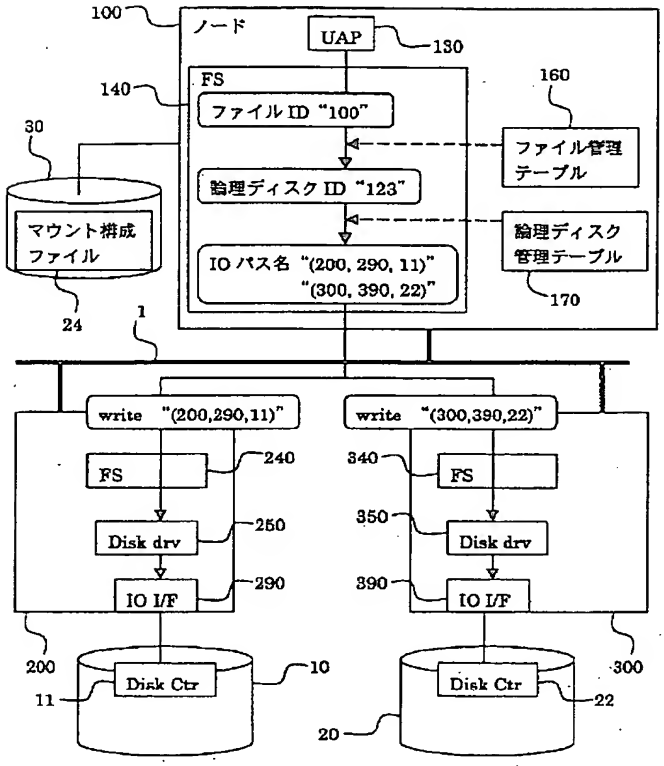
【図１４】

図 14



【図１５】

図15



【図１６】

図16

マウント構成ファイル２４

((ノード番号, IO I/F 番号, Disk Ctr 番号) 使用可否)　マウントポイント　⟵ コメント行

((200, 290, 11) available) ((300, 390, 22) available) /mnt

51　52　53　54　55

論理ディスク管理テーブル 270 (170)

271 (171) 論理ディスク ID	272 (172) ノード番号	273 (173) IO I/F 番号	274 (174) Disk Ctr 番号	275 (175) 状態フラグ	276 (176) ノード番号	277 (177) IO I/F 番号	278 (178) Disk Ctr 番号	279 (179) 状態フラグ
123	200	290	11	使用可	300	390	22	使用可

Ｆターム(参考) 5B014 HA09 HA13 HB01 HB26
5B018 GA10 HA40 KA11 MA12 QA01
5B065 CC01 EA12
5B082 EA01 FA05
5B083 AA08 BB03 CC04 CD11 EE08